Proximity-based Document Representation for Named Entity Retrieval

Desislava Petkova
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
petkova@cs.umass.edu

W. Bruce Croft
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
croft@cs.umass.edu

ABSTRACT

One aspect in which retrieving named entities is different from retrieving documents is that the items to be retrieved – persons, locations, organizations – are only indirectly described by documents throughout the collection. Much work has been dedicated to finding references to named entities, in particular to the problems of named entity extraction and disambiguation. However, just as important for retrieval performance is how these snippets of text are combined to build named entity representations.

We focus on the TREC expert search task where the goal is to identify people who are knowledgeable on a specific topic. Existing language modeling techniques for expert finding assume that terms and person entities are conditionally independent given a document. We present theoretical and experimental evidence that this simplifying assumption ignores information on how named entities relate to document content. To address this issue, we propose a new document representation which emphasizes text in proximity to entities and thus incorporates sequential information implicit in text. Our experiments demonstrate that the proposed model significantly improves retrieval performance. The main contribution of this work is an effective formal method for explicitly modeling the dependency between the named entities and terms which appear in a document.

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: H.3.1 Content analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms: Algorithms, Performance

Keywords: NE (named entity) retrieval, expert finding, language models, proximity kernels

1. INTRODUCTION

A named entity is a semantic category, a pointer to a real world entity such as a city, an organization, a movie, a book, or a historical event. Named entities are complex language features that are much richer in semantic content than most vocabulary words; they occur often in documents, particularly in news. Named entities have been shown useful for improving retrieval performance in a variety of tasks such as new event detection [9] and cross-language retrieval [11].

Named entities of particular interest are those referring to individuals – person entities. Recent work in retrieving people was motivated both by its practical importance and by the introduction of the Enterprise track in the Text REtrieval Conference [4]. The track provides a common platform for empirically comparing entity modeling techniques on expert search, i.e. the task of finding people who have skills and experience on a given topic. For example, given the query *"privacy on the web"* a relevant entity is someone who works on policies or technologies related to internet privacy. More specifically, the task involves finding evidence (documents or snippets of text) to support the claim of expertise as well as combining the evidence to estimate the probability of a person being expert given the text.

Automatic methods to rank person entities according to expertise can be applied for a variety of tasks: to help the user identify the most qualified person to contact, or find prospective collaborators for a project. Promising applications include an expert recommender system for enterprise search [17], automatically assigning conference submissions to reviewers according to their research interests [6] and matching job applications to potential employers [8].

We present a generative language modeling approach for entity finding that is based on estimating the joint distribution of words and entities. In particular, we focus on the dependency between the two (random) variables. Such dependency naturally exists in text because the words directly or indirectly describe the entities. Our contributions include

- Kernel-based document representation to incorporate positional information by fitting a multinomial density around named entity occurrences in a document (Section 3.1).
- Investigation of several aspects of named entity retrieval, including the assumption of query term independence, the effect of named entity extraction on retrieval effectiveness, and Dirichlet smoothing for entity representations (Sections 3.2, 3.3 and 3.4).

We evaluate the proposed model on the task of finding experts but the model is completely unstructured and domain-independent in that the knowledge, interests and expertise of candidate experts are not explicitly represented. Instead we estimate a probability distribution over vocabulary terms to describe the context in which a named entity appears.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'07, November 6–8, 2007, Lisboa, Portugal.
Copyright 2007 ACM 978-1-59593-803-9/07/0011 ...$5.00.

2. BACKGROUND AND RELATED WORK

Named entity retrieval is based on the assumption that terms co-occurring with a named entity in the same context describe it in a useful way.

Conrad *et al.* [3] used fixed-size window around proper names to automatically construct representations of people for retrieval as well as for finding statistically significant associations between entities. Raghavan *et al.* [14] formally stated this approach in the language modeling framework for retrieval and successfully applied this approach to a variety of tasks: fact-based question answering, classification into predefined categories, clustering and selecting keywords to describe the relationship between similar entities.

These studies are based on creating a pseudo-document (often called profile) by concatenating associated text segments. These virtual 'entity documents' can then be indexed and ranked directly given a query. Profiles can be significantly smaller in size than the original corpus, making this technique very efficient. Computationally, building profiles can be viewed as a form of clustering to reduce the dimensionality of the data and to produce compact entity representations. However, better space management comes at the price of losing information contained in individual documents such as document structure.

Recently, a family of probabilistic methods have been proposed which represent entities indirectly as a weighted mixture of documents from the collection [2, 1, 7]. These models use document retrieval techniques to estimate components of the entity models, in particular to weight the evidence using $p(t|d)$ and to assign the mixture parameters using $p(c|d)$ where t is a query, c is a candidate and d is a document. In contrast to the profile-based approach, the document-based approach preserves all information contained in the collection. When entities are modeled through a set of associated documents, their structure and high-level language features can be successfully used to improve performance [13].

Three models that have recently been successfully applied in the expert search task of the TREC Enterprise track: Cao *et al.* [2] propose a two-stage language model which combines a co-occurrence model to retrieve documents given a query, and a relevance model to find experts in those documents. Balog *et al.* [1] describe "Model 1" which directly represents the knowledge of an expert from associated documents, and "Model 2" which first locates documents on the topic and then finds the associated experts. Fang *et al.* [7] derive a generative probabilistic model from the probabilistic ranking principle and extend it with query expansion and nonuniform candidate priors.

3. MODELING NAMED ENTITIES

These models are instances of a document-driven generative language-modeling approach to rank named entities. Formally, given a set of documents $\mathcal{S} = \{d\}$ and a query topic $t \in \mathcal{V}^*$, we can rank candidate entities $c \in \mathcal{E}$ by the joint distribution $p(c, t)$ of candidates and terms. To achieve this, we treat the documents as independent random samples drawn from $p(c, t)$ and represent the joint as a weighted average of the document models.

$$p(c,t) = \sum_{d \in \mathcal{S}} p(c,t|d)p(d) \tag{1}$$

This general formulation can be factorized further to derive the expert finding models described above, as we show in Section 4. In this work, we focus on estimating $p(c, t|d)$, and in particular, representing the conditional dependence between the named entities and terms in a document. For simplicity we assume that $p(d)$ is uniform and thus can be ignored for the purpose of ranking entities. Alternatively, static (query-independent) ordering of the documents can be computed using a variety of techniques [15].

Next, without loss of generality, let

$$p(c,t|d) = p(t|c,d)p(c|d) \tag{2}$$

Thus we decompose the problem of named entity retrieval into two subproblems: named entity extraction to find support documents about a candidate – this corresponds to $p(c|d)$, and named entity ranking to model the distribution of terms in support documents – this corresponds to $p(t|c, d)$.

Finally, we point out that the document-based density estimation is based on the Relevance Model by Lavrenko *et al.* [10] for approximating the probability distribution of terms in the relevant class of information need I. The information need is represented by the set of query terms which are randomly sampled from the relevance model $p(\cdot|I)$. Assuming i.i.d. (independent and identically distributed) sampling, the joint distribution $p(w, I)$ for some vocabulary term $w \in \mathcal{V}$ is estimated from a finite set of document models M:

$$\begin{aligned} p(w,I) &= \sum_{M \in \mathcal{M}} p(w,t_1,\ldots,t_k|M)p(M) \\ &= \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} p(w|M)p(t_1,\ldots,t_k|M) \end{aligned}$$

In expert finding, the information need is a candidate expert c and the sampling space is the subset of documents in which a reference to c occurs, which we call a support set.

The essence of named entity modeling as defined in Eq. (1) is estimating the probability $p(c, t|d)$. If we make the simplifying assumption that words and candidates are independent given a document, then $p(c, t|d) = p(c|d)p(t|d)$ and the two components can be computed separately by treating c and t as queries and using a probabilistic retrieval method. However, conditional independence between c and t is a very strong assumption. Intuitively it ignores the relationship between words and named entities that appear in the same document.

Theoretically this assumption implies that a candidate is equally associated with all the topics discussed in a document. This certainly holds when the person is the author of the text but otherwise this might not hold, especially for long documents which are more susceptible to topic drift. A common example is a technical report written by a group of colleagues where each team member writes a section about her own work. Moreover, in the general case of retrieving named entities of arbitrary type, the notion of authorship is meaningless; therefore looking for author-document relationships would be a domain-specific solution.

However, estimating $p(t|c, d)$ poses a challenge: dependence between the terms and entities in a document violates the bag-of-words assumption. Therefore, a new document representation is required to capture this dependency explicitly. Motivated by previous work on leveraging positional information to improve retrieval performance [5, 12],

we propose such a document representation which is based on the proximity between occurrences of entities and terms.

3.1 Proximity kernels

First we consider the simplest case when the query is made up of a single term t. Bag-of-words is the document representation most often used in language modeling:

$$p(t|d) = \frac{1}{N}\sum_{i=1}^{N}\delta_d(i,t)$$

where $\delta_d : \mathbb{N} \times \mathcal{V} \to \{0,1\}$ is an indicator function such that

$$\delta_d(i,t) = \begin{cases} 1 & \text{if the term at position } i \text{ in } d \text{ is } t \\ 0 & \text{otherwise} \end{cases}$$

The constant N is the length of document d and it ensures normalization.

We propose document representation which uses positional information – the distance between query terms and candidate names – to weight the dependency between t and c. Our hypothesis is that the proximity of co-occurrence is helpful for estimating the strength of association between the c and t random variables. This is simpler than describing the nature of the relationship. But since the task is to find experts we implicitly label the relationship as *c is an expert on t* and we assume that the support is positive.

We formulate a candidate-centered document representation which considers the distance $|t-c|$ between the positions at which the term and the candidate occur:

$$p_k(t|c,d) = \frac{1}{Z}\sum_{i=1}^{N}\delta_d(i,t)k(t,c),\ Z = \sum_{i=1}^{N}k(t,c)$$

where k is a proximity-kernel function and Z is a normalizing constant which guarantees that p_k is a distribution.

Any non-uniform, non-increasing function can be converted into a proximity-based kernel. The simplest kernel is a constant function which assigns the same probability to each term in the document, thus ignoring any positional information. This degenerate case corresponds to the bag-of-words representation:

$$k(t,c) = \frac{1}{N}$$

Next we define three non-uniform kernel functions. They are presented graphically in Fig. 1.

1. Triangle kernel:

$$k(t,c) = 1 - \tan(\gamma)\cdot|t-c|$$

where γ is the angle between the leg of the triangle and the x-axis. Points outside this range are assigned zero probability. By increasing the angle we decrease the spread of the distribution.

2. Gaussian kernel:

$$k(t,c) = \mathcal{N}(t,c,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-(t-c)^2}{2\sigma^2}\right]$$

where the variance σ controls how quickly the curve tails off. All positions are assigned nonzero probability but we can vary the interval in which most of the probability mass is concentrated by tuning σ. Thus the variance can be interpreted as the distance within which we expect to find text that describes the entity in a useful and reliable way.

3. Step function:

$$k(t,c) = \sum_{j=1}^{m}\alpha_j \dot{I}_{A_j}(|t-c|),\ I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

where A is a sequence of m intervals with weight α and I_A is an indicator function. A step function is difficult to optimize because we need to specify the length and weight of each interval. In our experiments we keep the intervals fixed (their length increases in increments of fixed size) and we set the weight as a function of a Gaussian variable whose variance σ we vary to optimize the step function.

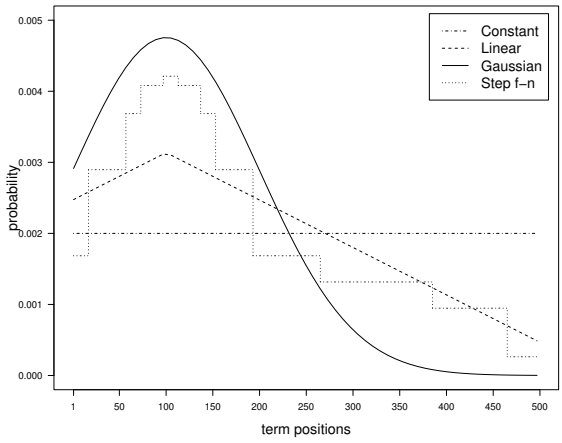

Figure 1: Proximity kernels. The constant function distributes probability mass uniformly across positions. The other three kernels "lift" the probability around the candidate occurrence, so that adjacent terms are assigned higher probability.

3.2 Term independence in entity models

Most queries contain multiple terms. In this case, the intuitive dependence between terms that exists in natural text is usually relaxed to make computation tractable. In Eq. (2) we estimate two types of models: c for candidates and d for documents, therefore we can make two independence assumptions leading to different sampling methods.

We can assume that terms are independent in c; then

$$p(c,t) \overset{\text{rank}}{=} \prod_{t_i \in t}\left\{\sum_{d \in \mathcal{S}} p(t_i|c,d)p(d|c)\right\} \tag{3}$$

Model 1 by Balog *at al.* [1] makes this independence assumption. The model is also referred to as the topic generation model with profile-based estimation by Fang *et al.* [7].

Alternatively, we can assume that terms are independent in d. In this case terms are exchangeable but not independent in c; then

$$p(c,t) \overset{\text{rank}}{=} \sum_{d \in \mathcal{S}}\left\{\prod_{t_i \in t} p(t_i|c,d)\right\}p(d|c) \tag{4}$$

Model 2 by Balog *et al.* and the candidate generation model by Fang *et al.* fall into this category.

It is obvious that the these two expressions are not equivalent mathematically since the product and the sum cannot

.....   Thanks in advance.   =====   XXXXXXX X XXXXXX, Ph.D.   Department of Electrical Engineering   XXXXXX University
Slide list: The Semantic Web   by XXXX XXXXXX   Slide: Digital Libraries and the Semantic Web   Formalized facilities for supporting ontology merging   .....

Table 1: Term independence matters when the document partially matches the query. Topic EX52 is *"ontology engineering"*. The first segment incorrectly provides support under the independence in c assumption because it matches "engineering". The second segment incorrectly provides no support under the independence in d assumption because it does not match both "ontology" and "engineering".

be interchanged. To give a probabilistic interpretation, we can regard $p(t|c,d)$ as a measure of how well document d supports the claim that c is an expert on t. If t is the entire query, then we expect to find all query terms in a support document, though not necessarily as a phrase. If t is a single query term, then we expect to find at least one query term in a support document. Table 1 illustrates the difference with a small example. It is important to note that both assumptions are easily violated.

The two independence assumptions imply different sampling frameworks as depicted in Fig. 2. The independence in d assumption implies the following: sample a document, then sample all query terms from the document model. To an extent this captures dependencies between query terms implicitly because a candidate is considered expert if we find many pieces of evidence which independently prove expertise. But this method is less flexible: only documents that match the whole query are considered to provide support.

The independence in c assumption implies the following sampling framework: for each query term sample a document, then sample the term from the document model. This method is less effective in capturing dependencies between terms because documents are not expected to be evidence for expertise on their own. Thus candidates are not penalized if the query terms never appear together in the same support document, although they might appear in many support documents. But this sampling is more flexible: it is less likely to overlook good documents which otherwise will be ignored because information is not always described in exactly the same terms (the problem of synonymy).

3.3 Finding entities in text

In Eq. (3) and (4) the conditional probability $p(d|c)$ can be viewed as a weight function which indicates how well a particular document describes a candidate since d contributes to the final estimate of c in proportion to $p(d|c)$. How well this function is estimated can have significant effect on performance as it determines how successfully we retrieve information about the candidates.

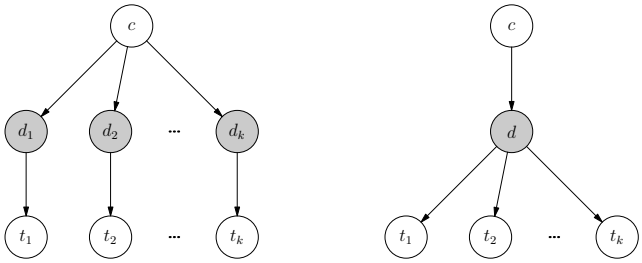

Figure 2: Under independence in c, a different document d_i is sampled for each query term t_i (left). Under independence in d, first a document d is sampled and then all query terms are sampled from the same document (right).

candidate-0001 Dan Brickley danbriw3.org	dan brickley; brickley, dan d. brickley ; brickley, d. daniel brickley danbri@w3.org; danbri daniel.brickley@bristol.ac.uk
candidate-0070 Björn Höhrmann derhoermi@gmx.net	bjorn hohrmann bjoern hoehrmann bjrn hhrmann derhoermi@gmx.net

Table 2: TREC provides a short description of each candidate expert (left column). Actual references are often not consistent with that description. Examples include combinations of names and initials, alternative emails and spelling errors due to different character encodings (right column).

Balog *et al.* [1] used several rules to match proper names to candidate IDs. Document weights can also be estimated by treating c as a query and using a query-likelihood retrieval method to score documents; then $p(d|c)$ can be computed by normalizing $p(c|d)$ over $d \in \mathcal{S}$. Formulating this candidate query poses challenges of its own because of the named entity disambiguation problem: people are not always referred to with their exact full name (Table 2).

We emphasize that even though $p(c|d)$ is a probability distribution, it should not be viewed as a statistical approach for extracting named entities. It is still a rule-based method because we need to specify what constitutes a candidate occurrence (e.g. full name or last name) so that the term frequency function can be computed. Using the Indri search engine [16] which provides a variety of positional and logical operators we can devise some quite complicated matching rules for detecting entity occurrences (Table 3).

3.4 Smoothing entity models

When building a document model in the language modeling framework for IR, we have a finite sample to estimate the density. In many cases, particularly for short documents, maximum likelihood estimation can lead to poor results by assigning zero probability to terms that do not appear in the document. Entity modeling also faces the problem of underestimating the probability of non-occurring terms.

The goal of smoothing is to add a small background probability, so that the mass is redistributed and any event has a nonzero probability to occur [18]. Certainly smoothing the

MATCH1	`#1( danbri w3org )`	Matches email as an exact phrase
MATCH2	`#od( dan brickley )`	Matches first or last name; both names must appear in the document
MATCH3	`#1( dan brickley )`	Matches full name as an exact phrase
MATCH4	`#od2( dan brickley )`	Matches full name as an ordered bigram; first and last names can be separated by at most one term, e.g. middle initial
MATCH5	`#weight( 0.8 #uw2( dan brickley ) 0.2 #1( danbri w3org ) )`	Mixture of full name and email address

Table 3: Rules for matching candidate occurrences (here defined using Indri proximity operators) have limited descriptive power. For example, neither rule will detect all occurrences of candidate 0001 because it is not apparent from the specifications that "Dan" is short for "Daniel".

document models before combining them to represent entities will have the effect of assigning nonzero probability to any vocabulary term $t \in \mathcal{V}$. But this can introduce noise from incorrect associations. Since the sample space is the entire collection $\mathcal{S}$ – much larger than an average document, though still finite – we could expect that the observed joint distribution is very close to the true distribution $p(c,t)$.

Consider the following example: Query $EX62$ is "mereology". Candidate 0002 is not an expert on this topic but is the most frequent candidate appearing in 10567 documents. There are 18 documents which contain the term mereology but candidate 0002 does not occur in any of these. However, if the document models are smoothed the combined set of 10585 documents would be considered to provide support, however small, to the claim that 0002 is an expert. Although the background probabilities are small, they accumulate in the final estimate when summing over a large set of smoothed documents.

To avoid over-smoothing we formulate Dirichlet smoothing for entity representations as follows:

$$p(t|c) = \frac{l_c}{l_c+\mu} p_{ml}(t|c) + \left(1 - \frac{l_c}{l_c+\mu}\right) p(t|\mathcal{E}) \qquad (5)$$

where $\mathcal{E} = \{c\}$ is the set of candidate experts and l_c is the number of documents which are associated with candidate c and thus contribute positively to the corresponding representation. Smoothing can be viewed as profile-length normalization to prevent favoring candidates who occur in many documents and therefore have more chances to co-occur with query terms.

4. COMPARISON OF LANGUAGE MODELS FOR FINDING EXPERTS

In this section, we derive several language models proposed for expert finding from the general document-driven framework as given by Eq. (1). Our analysis shows that the models are similar – in fact, probabilistically equivalent – since they try to estimate the same underlying probability distribution. We highlight the simplifying assumptions that each model makes in order to give a theoretical explanation for the differences in performance. As stated in Eq. (1), the goal is to estimate the joint distribution $p(c,t)$. The joint can be factored in two equivalent ways using Bayes rule which states $\Pr(a,b) = \Pr(b|a)\Pr(a) = \Pr(a|b)\Pr(b)$ for two stochastic events a and b.

Using Bayes rule we can condition c on t and d:

$$\begin{aligned}
\sum_d p(c,t|d)p(d) &= \sum_d p(c|t,d)p(t|d)p(d) \\
&= \sum_d p(c|t,d)p(d|t)p(t) \\
&\stackrel{\text{rank}}{=} \sum_d p(c|t,d)p(d|t) \qquad (6) \\
&\stackrel{\text{rank}}{=} \sum_d p(c|d)p(d|t) \qquad (7)
\end{aligned}$$

To derive Eq. (6) we assume that the prior distribution over t is uniform. This gives the two-stage model proposed by Cao *et al.* If we also assume that c and t are conditionally independent given d, we get Eq. (7) which is the candidate generation model proposed by Fang *et al.* [7]. Note that while $p(t)$ is assumed uniform, the distribution over candidates $p(c)$ is implicitly nonuniform.

Alternatively, we can condition t on c and d:

$$\begin{aligned}
\sum_d p(c,t|d)p(d) &= \sum_d p(t|c,d)p(c|d)p(d) \\
&= \sum_d p(t|c,d)p(d|c)p(c) \\
&\stackrel{\text{rank}}{=} \sum_d p(t|c,d)p(d|c) \qquad (8) \\
&\stackrel{\text{rank}}{=} \sum_d p(t|d)p(d|c) \qquad (9)
\end{aligned}$$

To derive Eq. (8) we assume that the prior distribution over c is uniform. If we also assume that c and t are conditionally independent given d, we get Eq. (9) which is Model 2 proposed by Balog *et al.* [1]. In contrast to the previous derivation, $p(c)$ is assumed uniform while the distribution over topics is implicitly nonuniform. However, the latter condition is irrelevant because each topic is evaluated independently of the others.

The above derivations show that the models are probabilistically equivalent because they estimate the joint distribution $p(c,t)$ as the sum $\sum_d p(c,t|d)$ over the sample of collection documents. However, the models do not have the same performance in practice because they make different assumptions about the prior distribution over candidates and the term independence in candidate representations.

On one hand, we can choose $p(t)$ or $p(c)$ to be uniform. Assuming that $p(t)$ is uniform does not affect the ranking since this distribution does not depend on the candidates. But the candidate prior $p(c)$ can significantly influence performance. If a particular topic is difficult and the system cannot find enough evidence to decide who is an expert, it can use prior knowledge to rank the candidates.

On the other hand, we can assume conditional independence between c and d. As we already discussed in Section 3.1, intuitively this assumption is wrong because it ignores the relationship between the words and named entities which appear in a particular document. Therefore we expect that

both Model 2 and the candidate generation model to perform worse than our kernel-based entity retrieval model:

$$p(t,c) = \prod_{t_i \in t} \left\{ \frac{l_c}{l_c + \mu} \sum_{d \in S} p_k(t_i|c,d) p(d|c) + \left(1 - \frac{l_c}{l_c + \mu}\right) p(t_i|\mathcal{E}) \right\}^{w(t_i)}$$

where $w(t_i)$ is the relative weight of term t_i in the query model t. We make the independence in c assumption and we estimate $p(t_i|c,d)$ for each term t_i independently of other query terms and their occurrences (or lack thereof) in document d. We compute the proximity of a term and a named entity according to kernel k.

5. EXPERIMENTS

We run experiments on the dataset used in the expert search task of the TREC 2006 Enterprise track. It includes the W3C corpus, a list of 1092 candidate experts, and a set of 49 queries. The W3C collection is crawled from the World Wide Web Consortium web site and is quite heterogeneous in document types. In our experiments we use only the emails and web pages, 244369 documents in total.

We design our experiments to supplement the theoretical discussions. (Corresponding theoretical sections are references in parentheses.)

1. Named entity extraction component: We look at how the accuracy of named entity extraction – estimating $p(c|d)$ – affects retrieval effectiveness.

2. Query term independence: We compare two term independence assumptions with respect to their ability to find support documents (Section 3.2).

3. Smoothing: We compare smoothing document models individually vs. smoothing candidate representations after combining maximum likelihood models of documents (Section 3.4).

4. Proximity-kernel document representation: We test our hypothesis that directly modeling the conditional dependency between terms and entities can improve named entity retrieval (Section 3.1).

5. Uniformity of $p(c)$: We investigate whether the importance of prior knowledge about the entities changes as the entity representations improve (Section 3.3).

Implementation is based on the Indri search engine in the Lemur toolkit.[1] Indri integrates Bayes net retrieval model with formal statistical techniques for modeling relevance [16]. The Bayes net representation of an information need allows us to formulate richly structured queries, including phrase matching which we use to find occurrences of proper names. This functionality is combined with relevance estimation based on smoothed language models.

5.1 Finding entities in text

We are given a list of candidate experts with full name and an email address as input, and we assume that the list contains accurate and complete information. This simplifies the problem of identifying person entities in the text. Although designing better named entity extraction techniques is not the focus of this work, we believe that the quality of named entity extraction influences the performance of a named entity retrieval system.

People are not always referred to by their exact full names, especially in emails, and this makes identifying occurrences of person entities very challenging. In our first set of experiments we compare rule-based entity matching with named entity resolution. For rule-based person matching we use the Indri operators defined in Table 3. As an instance of a more advanced named entity extraction technique, we use annotations of candidate occurrences provided by Jianhan Zhu to participants in the TREC Enterprise track [19]. In this preprocessed version of the W3C dataset candidates are recognized by full name, name variations, email addresses, user ID, etc., using the Aho-Corasick matching algorithm.

Comparison of rule-based methods and name resolution shows that while finding candidate occurrences is essential for good performance, finding all occurrences is not (Table 4). Extraction performance measures are defined as follows: *num_occ* is total number of occurrences; *num_ret* is number of candidates with at least one occurrence; *top_k_50* is the number of most frequent entities that account for 50% of all occurrences; *mean* and *median* is the average and the median number of occurrences. Retrieval performance measures are defined as follows: *mAP* is mean average precision; *R-prec* is precision after all relevant items are retrieved; *MRR* is mean reciprocal rank of the top relevant item; *P5* and *P10* is precision at rank 5 and rank 10.

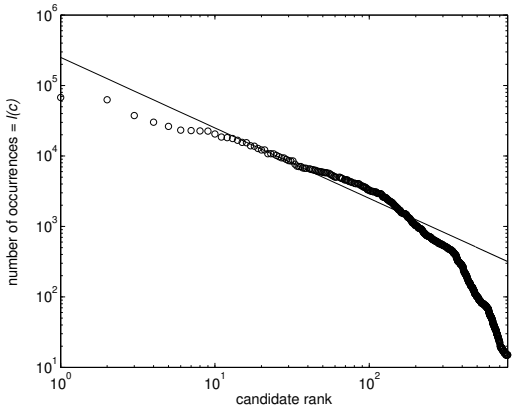

Figure 3: Candidate occurrences follow Zipf's law. Number of occurrences is plotted as a function of frequency rank (in loglog scale). The distribution predicted by Zipf (the straight line) is not a good fit for the tail of the observed distribution, i.e. there is not enough information about rare candidates.

As expected, name resolution is far more successful in identifying person entities in text: it finds more than 1,200,000 occurrences, while the matching rules find no more than 400,000. Name resolution also picks up entities that any matching rule fails to identify. (We do not have "truth" labels, therefore some of these occurrences might be false positives.) However, in terms of performance, the improvement is small and not statistically significant compared the most elaborate matching rule. The results indicate that

[1] http://lemurproject.org/

Extraction method	*num_occ*	*num_ret*	*top_k_50*	*mean*	*median*	*mAP*	*R-prec*	*MRR*	*P5*	*P10*
`MATCH1`	60,919	407	19	55.8	0	0.3819	0.4188	0.8605	0.6571	0.5653
`MATCH2`	329,154 (+440%)	782	39	301.4	14	**0.5133** (+34%)	**0.5296** (+26%)	0.8503 (-1.2%)	0.7102 (+8.1%)	0.6306 (+12%)
`MATCH3`	277,214 (-15.8%)	690	35	253.8	6	**0.5374** (+4.7%)	0.5471 (+3.3%)	0.8405 (-1.2%)	0.7388 (+4.0%)	0.6490 (+2.9%)
`MATCH4`	299,235 (+7.9%)	700	35	274.0	7	0.5508 (+2.5%)	0.5565 (+1.7%)	0.8656 (+3.0%)	0.7469 (+1.1%)	0.6633 (+2.2%)
`MATCH5`	360,154 (+20.4%)	710	34	329.8	8	0.5598 (+1.6%)	0.5660 (+1.7%)	0.8520 (-1.6%)	0.7388 (-1.1%)	0.6694 (+0.4%)
Name resolution	1,170,258 (+225%)	787	30	1071.7	25	0.5739 (+2.5%)	0.5595 (-1.1%)	0.8946 (+5.0%)	0.7347 (-0.6%)	0.6694 (0.0%)

Table 4: Improving entity extraction has diminishing returns. A name resolution technique finds three times as many occurrences as the best matching rule but the improvement in performance is small. Bold indicates statistical significance ($p < 0.05$ with Wilcoxon signed rank test) over the previous extraction method. *[Extraction methods are ordered according to implied complexity.]*

it is frequent candidates who benefit mostly from applying increasingly sophisticated entity extraction techniques. In fact, since employees participate to various degrees in company interactions, candidate occurrences follow a Zipf distribution, where a small number of people have a large number of occurrences and the majority of people have a small number of occurrences (Fig. 3). Therefore, most new occurrences belong to candidates for whom many documents have already been found.

Another issue to consider is that name resolution introduces noise from incorrect partial matchings. In this respect, advanced named entity extraction techniques such as co-referencing and name disambiguation can provide more accurate occurrence information.

5.2 Term independence in entity models

As discussed in Section 3.2, we can make two term independence assumptions: independence in c (Eq. 3) or independence in d (Eq. 4). They correspond to different sampling methods and induce different support documents. The set of documents over which the relevance score for a candidate expert is computed is always the same: documents which contain at least one reference to the candidate, denoted by $\mathcal{F}_c$. A supporting document $d \in \mathcal{F}_c$ provides evidence for expertise on a given topic t, i.e. d gives c a high probability of being an expert on t.

We compare the two sampling methods empirically by repeatedly expanding the query. Our goal is to understand how the retrieval effectiveness improves as we build more accurate models of the information need. As an example of a model that makes the independence in d assumption we take the candidate generation model. As an example of a model which makes the independence in c assumption we take our kernel-based model. *[We use the constant kernel in order to make a fair comparison.]*

Many techniques have been developed for building complex query models which capture high-level language features. Typically they expand the query by adding terms automatically estimated to be related to the topic. In this experiment we apply two expansion techniques: the Term Dependency Model [12] and the Relevance Model [10] for pseudo relevance feedback. Both assign each new term a weight $w(t_i)$ which indicates its relevance in the expanded query model t^*.

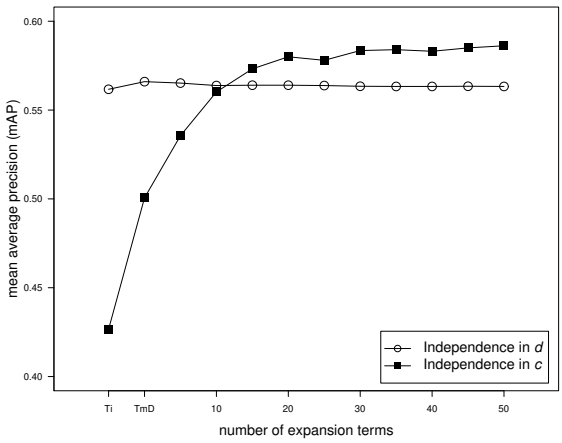

Figure 4: The two independence assumptions behave differently under query expansion. For *Ti* we use the `<title>` field only; for *TmD* we add the Term Dependency model; then we incrementally add 5 terms from pseudo relevance feedback.

We present query-expansion results in Fig. 4. As expected, the independence in d assumption gives significantly better performance when only the original keyword query is used. However, retrieval performance under the independence in c assumption consistently improves as the query model becomes more precise, while performance under the

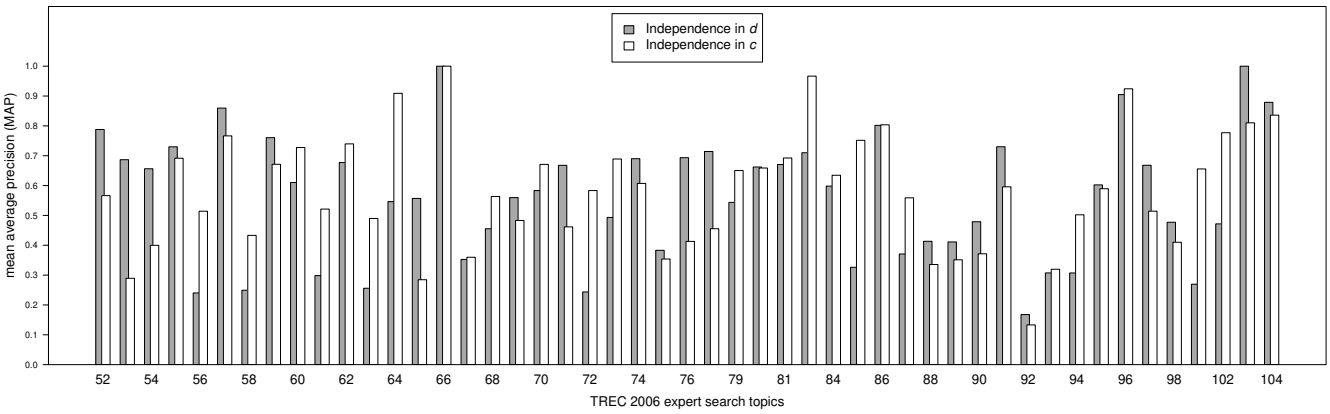

Figure 5: Comparison of the two independence assumptions over the TREC 2006 set of 49 queries.

independence in d assumption gets virtually no improvement from query expansion. With only 25 terms added, independence in c has a slight (though not statistically significant) advantage. Recall that independence in d is the stronger assumption: all query terms should occur in a support document; independence in c is the weaker assumption: at least one query term should occur in a support document.

Our hypothesis for this behavior is that independence in c is more flexible in finding support documents. Intuitively, a support document should provide proof for expertise on its own. This assumption underlies the independence in d sampling. However, the initial query might fail to capture some support documents, for example when a document does not contain a query term and instead contains a different term with similar meaning (synonym). In this case, query expansion can help by specifying related terms for the same information need. But under the assumption that terms are interchangeable, we want to see all query terms – both initial and new ones. This requirement is very restrictive. Under the assumption that terms are independent, we want to see any query term – initial or new one. Thus we can find new support documents, as opposed to measure the relevance of existing support documents better.

A detailed comparison of the two independence assumptions in Fig. 5 shows that there are a few queries (topics 1-3, 14, 25-26) for which the independence in d assumption gives substantially better results. We suspect that the independence in c assumption underperforms when query expansion fails and picks unrelated terms.

5.3 Smoothing entity models

In the language modeling framework, two smoothing techniques are usually compared – Jelinek-Mercer and Dirichlet. In view of the fact that entities are modeled as subsets of documents, we believe it is more important to compare smoothing each document individually versus smoothing the candidate model after maximum likelihood (not smoothed) document models are combined.

In Section 3.4 we proposed an entity smoothing technique similar to Dirichlet document smoothing in order to reduce noise from incorrect associations. (This occurs when a document is estimated to provide support for expertise when in fact it does not.) To smooth a candidate model, we use maximum likelihood estimates for documents in the profile set $\mathcal{F}_c$ and then interpolate the document mixture which represents the candidate with a background model.

Comparison for different values of the Dirichlet parameter μ shows that smoothing the candidate models directly is significantly more effective than smoothing the document models (Fig. 6). Essentially this method smooths the candidate representations once rather than again and again for each document in the profile set.

This experiment also shows that although techniques for ranking documents can be applied to retrieve named entities, the problem-specific aspects cannot be ignored. For example, in document retrieval there is no ambiguity in matching terms and documents are modeled independently. But in entity modeling identifying candidates is hard and there is interaction between documents as the information they contain is combined to produce the entity ranking.

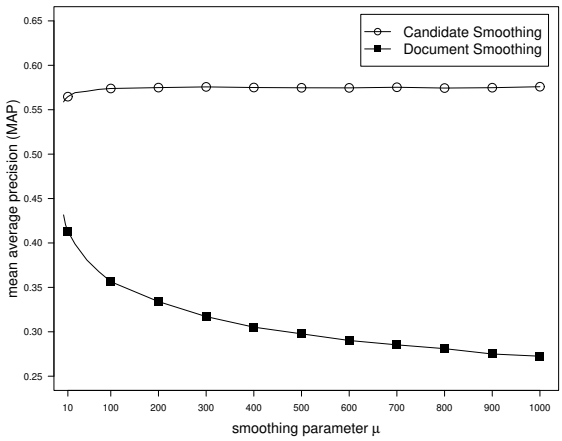

Figure 6: We can smooth either the document models or the candidate models. The second approach is more robust to noise from incorrect associations.

5.4 Proximity kernels

In Section 3.1 we proposed a proximity-based kernel representation where a document multinomial is "built around" candidate occurrences, rather than spread uniformly across the document as in the bag-of-words representation.

A document can contain multiple references to a particular candidate. In this case we place a multinomial at each occurrence. If a candidate appears twice in a document, two multinomials are estimated but the distributions are different since they are centered at different locations. If the two occurrences are close together the terms that appear in between are boosted by both multinomials.

Each nonuniform kernel improves performance on all measures (Table 5). The kernel parameter controls how the probability mass is distributed across term positions: increasing σ or decreasing γ either have the effect of making the distribution more uniform. Performance sensitivity experiments are reported in Fig. 7. The Gaussian kernel is least sensitive to parameter setting and its performance is near its optimal for a wide parameter range. The shape of the performance curve for the step function is very similar to that of the Gaussian kernel. This can be explain by the fact that we use the Gaussian distribution to automatically set weights for the intervals of the step function.

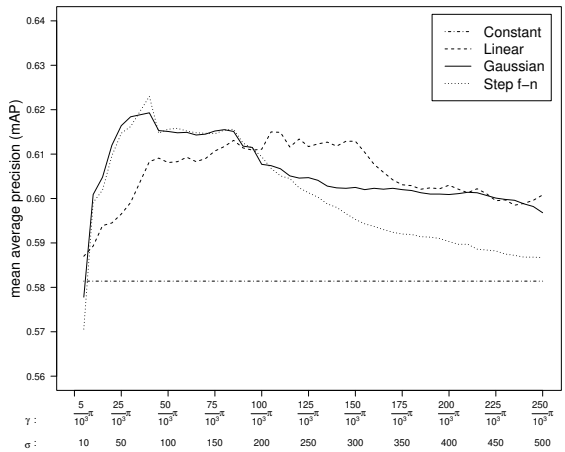

Figure 7: Parameter sweep for the kernel functions. As the γ and σ parameter increase, the probability mass is more evenly spread across the length of the documents, peaks at candidate occurrences are smoothed out and performance converges to the baseline bag of words representation.

Our kernel-based approach is similar to the window-based two-stage model for expert finding [19]. However, window-based retrieval is heuristic and it involves many parameters: number of windows, length and weight for each window. Due to the necessity to tune all these parameters the method is prone to overfitting. Our formal method includes window-based estimation as a special case (the step function). More importantly, two of the kernel functions we define – Gaussian and Linear – have only one parameter to optimize. Between those, the Gaussian kernel performs better and it is also robust with respect to its variance parameter.

Kernel	parameter	*mAP*	*MRR*	*P5*	*P10*
Constant		0.5814	0.9150	0.7633	0.6776
Triangle	γ=π/10	**0.6111**	0.9439	**0.8041**	0.7102
Gaussian	σ=80	**0.6193**	0.9541	**0.8163**	0.7041
Step f-n	σ=80	**0.6230**	0.9541	**0.8163**	0.6980

Table 5: Performance of the kernel-based representation on the expert finding task. Bold indicates statistical significance over the baseline (the constant function). Differences between the three nonuniform kernels are not statistically significant.

	c and t independent given d	c and t dependent given d
$p(c)$ is uniform	Topic generation model, Model 2	Two-stage model, Kernel-based model
$p(c)$ is nonuniform	Candidate generation model	

Table 6: Although the language models estimate the same joint probability $p(c,t)$, they make different assumptions, hence the difference in mean average precision (mAP).

5.5 Prior distribution over entities

As we noted in Section 4, an important difference between the different models for expert finding is whether they assume that the distribution $p(c)$ over candidate experts is uniform or not. Our classification of the models is summarized in Table 6.

In this experiment we evaluate the importance of prior knowledge about the candidates on retrieval performance. A variety of techniques for estimating the prior distribution $p(\cdot)$ can be devised, e.g. PageRank for entities, but we use a straightforward statistical method to compute $p(c)$ directly from the data.

$$p(c) = \sum_d p(c,d) = \frac{1}{|\mathcal{F}_c|} \sum_{d \in \mathcal{S}} p(c|d)$$

We find that direct evidence from text outperforms heuristic evidence of prior knowledge as the candidate representations get better (Table 7).

6. CONCLUSION AND FURTHER WORK

Web pages	Emails	Gaussian kernel	Uniform $p(c)$	Nonuniform $p(c)$
×			0.2944	0.4084
×	×		0.5850	0.5541
×	×	×	0.6193	0.5976

Table 7: Performance with nonuniform prior on candidate expertise.

Expert finding method	*mAP*	*R-prec*	*MRR*	*P5*	*P10*
Model 2	0.3571	0.3832	0.5296	0.4327	0.4184
Cand. generation	0.5617	0.5583	0.8929	0.7551	0.6571
Proxim. kernels (Gaussian k)	0.6193	0.5942	0.9541	0.8163	0.7041

Table 8: Performance of three language modeling approaches on the expert search task.

We focused on the problem of building language models of person entities and ranking them in terms of expertise on a given topic. We found that named entity extraction is not the primary factor for performance when working with a large collection and given reasonable identification rules. We suspect that this is the case because the simple rules match a sufficient portion of the actual number of occurrences.

We also analyzed the performance of two term independence assumptions which correspond to different sampling methods to aggregate evidence for expertise over many documents. We formally analyzed the behavior of the two sampling methods; we hypothesized that assuming terms are independent rather than dependent but interchangeable would lead to finding more documents to support expertise. Our hypothesis was confirmed by experimental results which show that the weaker assumption improves significantly when query expansion is applied.

We analyzed how smoothing influences performance, which motivated us to propose Dirichlet smoothing for entity representations. This reduces noise from smoothing each document in a potentially huge collection. We also proposed a proximity-kernel document representation to explicitly model the dependency between terms and entities. Our language model, which combines these features, effectively discovers evidence for expertise in a heterogeneous collection and has excellent performance on the TREC 2006 expert search task.

Finally, we presented a unified analysis of the probabilistic semantics underlying previous language modeling approaches for expert finding. We argued that they are equivalent probabilistically as they estimate the joint probability $p(c, t)$ but different statistically as they factorize the joint in different ways and make different assumptions.

Our model is very general because it associates named entities with text segments without relying on document structure or even paragraph or sentence decomposition. However, in special cases such as emails and scientific papers, knowledge about document structure can be applied to extract relationships, e.g. who is the author and who is the intended audience. As future work we plan to incorporate this additional source of information in our model for finding experts. We would also like to apply our kernel-based model to retrieve named entities other than persons to test the claim of its generalizability.

Another line of research is to use incrementally constructed entity representations to guide named entity extraction. We can simultaneously discover partial name matchings of an entity and build its representation; new information has to be consistent with the current incomplete representation before being added. Significant differences (e.g. in terms of KL divergence) might indicate the discovery of a new entity.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract #NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2006.

[2] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at Enterprise track of TREC 2005. In *Proceedings of the Text REtrieval Conference (TREC)*, 2005.

[3] J. G. Conrad and M. H. Utt. A system for discovering relationships by feature extraction from text databases. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 1994.

[4] N. Craswell, A. de Vries, and I. Soboroff. Overview of the TREC 2005 enterprise track. In *Proceedings of the Text REtrieval Conference (TREC)*, 2005.

[5] O. de Kretser and A. Moffat. Effective document presentation with a locality-based similarity heuristic. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 1999.

[6] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 1992.

[7] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2007.

[8] B. J. Jansen, K. J. Jansen, and A. Spink. Using the Web to look for work: Implications for online job seeking and recruiting. *Journal of Internet Research*, 15(1), 2005.

[9] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2004.

[10] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2001.

[11] T. Mandl and C. Womser-Hacker. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *Proceedings of the ACM Symposium on Applied Computing*, 2005.

[12] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2005.

[13] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of the IEEE Conference on Tools with Artificial Intelligence (ICTAI)*, 2006.

[14] H. Raghavan, J. Allan, and A. McCallum. An exploration of entity models, collective classification and relation description. In *Proceedings of the ACM SIGKDD Workshop on Link Analysis and Group Detection*, 2004.

[15] M. Richardson, A. Prakash, and E. Brill. Beyond PageRank: machine learning for static ranking. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2006.

[16] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. Technical report, Department of Computer Science, University of Massachusetts, Amherst, 2005.

[17] D. Yimam-Seid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis and the DEMOIR approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1), 2003.

[18] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2001.

[19] J. Zhu. Open University at TREC 2006 enterprise track expert search task. In *Proceedings of the Text REtrieval Conference (TREC)*, 2006.