A Ranking Framework for Entity Oriented Search using Markov Random Fields

Hadas Raviv, David Carmel
IBM Research lab, Haifa 31905, Israel
{hadasra,carmel}@il.ibm.com

Oren Kurland
Faculty of Industrial Engineering and Management
Technion, Haifa 32000, Israel
kurland@ie.technion.ac.il

ABSTRACT

In this work we present a general model for entity ranking that is based on the *Markov Random Field* approach for modeling various types of dependencies between the query and the entity. We show that this model actually extends existing approaches for entity ranking while aggregating all pieces of relevance evidences in a unified way. We evaluated the performance of our model using the INEX datasets. Our results show that our ranking model significantly outperforms leading INEX systems in the tracks of 2007 and 2008, and is equivalent to the best results achieved in the 2009 track.

1. INTRODUCTION

Answering the user's information need is one of the fundamental tasks of information retrieval (IR). The basic adhoc retrieval task has been recently extended to answer user queries by various types of entities. This extension follows the observation that for many user queries, named entities such as people, organizations, locations and products, are more suitable for query satisfaction than full documents such as web-pages, or scientific papers [16].

One of the questions that the emerging research field of *Entities oriented Search (EoS)* tries to deal with is how to rank entities in response to a given query [14]. The main distinction between entity ranking and document ranking is the characteristics of the items needed to be ranked. A document is a well defined object in most data collections. An entity, on the other hand, is an abstract thing characterized by its properties such as its name (which is often not unique, e.g., George Bush, George W. Bush), its type (which can be very general, e.g., a person, or more specific, e.g., a king), and many more. Moreover, an entity can be described or mentioned by many documents, or can be represented by a specific document (e.g. the entity's Wikipedia page, or the entity's homepage). The complexity in entity representation make *entities ranking* a challenging task.

In this paper we propose a general model for ranking entities in response to a given query. This model integrates various entity properties so as to estimate the entity's relevance to the query. Our model is based on the *Markov Random Field (MRF)* approach [18] for modeling the dependencies between the query and the entity. We show that this model actually extends existing approaches for entity ranking while aggregating all pieces of evidences about the relevance of the entity to the query in a unified way. We evaluate the performance of our model using the INEX *entity ranking track* datasets [10, 11, 13]. Specifically, we compare our model's performance with that of other ranking models that were proposed by the track participants over the years. Our results show that our approach significantly outperforms leading INEX systems in the tracks of 2007 and 2008, and is equivalent to the best results achieved in the 2009 track.

JIWES'12 August 16, 2012, Portland, Oragon, USA.
ACM 978-1-4503-1472-5/12/08.

2. RELATED WORK

In recent years several *entities retrieval* tasks have been explored. The *expert search task*, defined by the TREC *enterprise track*, took place throughout 2005-2008 and focused on searching for employees in the enterprise who are the most knowledgeable about a given topic [9]. INEX, the initiative for evaluation of XML retrieval, ran the *entity ranking track* between 2007 and 2009. The goal of that track was to retrieve entities that are the most relevant to a topic, described in natural language, from the English Wikipedia data collection. Candidate entities were restricted to items having their own Wikipedia article and the types of entities to retrieve (the entity target types) were explicitly defined by the corresponding Wikipedia categories [10, 11, 13]. The TREC *entity track* launched and ran throughout 2009-2011 aiming to perform an entity oriented search over the web. The goal of the *entity related finding task* of this track was to provide a ranking of entities that are related to an input entity according to a specific relationship type. Retrieved entities were represented by their web homepage [6].

Those various tasks differ in several dimensions, among which are the way the information need was specified, the entity target type, the collection used for retrieval, and more. Still, facing the general challenge of entity ranking led to the development of some common approaches which are described below.

Basic Retrieval approaches. Two main approaches exist for generating a ranked list of entities for a given query: "profile based approach" and "voting approach" (also referrd as "Model 1" and "Model 2" in Balog et al. [2]).

In the "profile based approach", entities are extracted from a given collection using various entity extraction tools. Then, a document representing each entity is constructed, for example, by concatenating the passages in which the entity appears. Finally, a standard document ranking method (e.g., BM25, LM, TF.IDF) is used to rank the created documents

with respect to the given query [1, 15]. The main challenges imposed by this approach are the creation of the representative entity document as well as pre-processing of the data collection in order to extract all entities.

In the "voting approach", the query is used to retrieve an initial list of documents from the collection [20, 2]. Then, entities are extracted from these documents using various extraction tools. Finally, estimators devise to estimate the relation of the extracted entities to the query are integrated to create a ranked list of entities. A popular estimator is one that sums the number of occurrences of the entity over the top scored documents, while considering the document score [2]. Another estimator takes into account the proximity of the entity to the query terms in the documents [19].

Filtering Approaches. The entity type is one of its most important property. An initial list of retrieved entities can be filtered, or re-ranked, according to the target entity type as defined by the query. The most detailed information regarding the target entity type was given by the INEX *entity ranking task*. Most participants made use of the Wikipedia categories tree structure to evaluate the relationship between the entity type and the target type [12]. Another approach modeled the target type as well as the entity type (the Wikipedia categories of the entity page) as a probability distribution. The similarity between the two distributions was taken into account by the ranking model [3].

The TREC *entity related finding task* in 2008-2009 limited the target entity types to "person", "product" and "organization" (the type "location" was added in 2010) [6, 5]. A recent filtering approach extracted the target type from the query narrative using NLP tools [21]. Seed entities of that type were retrieved from the web to represent this target type, and the similarity between them and the candidate entity was used for filtering.

In 2011, the TREC *entity related finding task* canceled the previous restrictions on the entity target types and required that target types should be inferred directly from the topic narrative [7]. In one approach for addressing the new task, the language model of pages defining the target type was compared to that of pages containing the entity to compute the overall similarity [8].

3. MRF FOR ENTITY RETRIEVAL

A Markov Random Field (MRF) is a graphical model in which the joint distribution over a set of random variables is represented using an undirected graph G. The graph nodes represent the random variables and the graph edges represent the dependence semantics between them.

A MRF based model for modeling term dependencies for ad hoc retrieval tasks was introduced by Metzler and Croft [18]. According to this model, the graph G is composed of a document node and query terms nodes. Different dependence assumptions between the query terms and the document are modeled by different edge configurations in the graph. Documents are ranked by their probability of relevance to the query, estimated by:

$$Pr\left(D|Q\right) \stackrel{def}{=} \sum_{c \in C(G)} \lambda_c f\left(c\right);$$

D is a document; $Q = q_1 ... q_n$ is the query; $C(G)$ is the set of cliques in G; $f\left(c\right)$ is an energy function over clique configuration (also termed *feature function*); λ_c is the relative weight given to that particular clique.

Our unified model for entity retrieval estimates the relationship between the query and the entity using the MRF ranking model. In the model, an entity is characterized by three properties; 1) a descriptive document E_D (given or constructed automatically), 2) the entity type E_T (given or inferred), and 3) the entity name E_N (or a list of equivalent entity names). Analogously to using MRF for document retrieval, we construct three undirected graphs, G_D, G_T, G_N, for modeling the joint distribution of the query with E_D, E_T, and E_N, respectively. The graphs can model different types of term dependencies, as suggested by Metzler & Croft [18]; full independence (FI), sequential dependence (SD) and full dependence (FD). Figure 1 presents the three graphs of our model each of which represents a full dependence model.

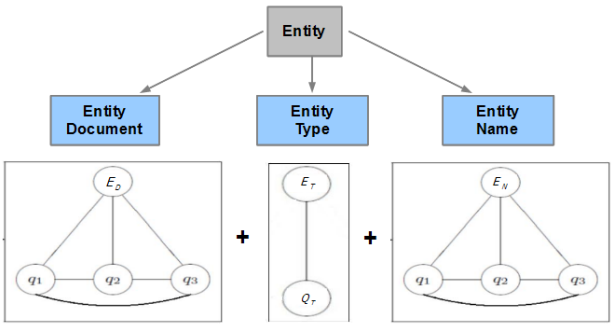

Figure 1: (Left) G_D - the joint distribution of the entity document with the query terms. (Middle) G_T - the joint distribution of the entity type with the query target type. (Right) G_N - the joint distribution of the entity name with the query terms.

The score of each of the entity properties, P ($\in \{D, T, N\}$) is estimated by $Pr(E_P|Q) = \sum_{c \in C(G_P)} \lambda_c f(c)$. The final retrieval score of an entity E is estimated by a linear aggregation of the scores of its three properties:

$$Pr(E|Q) \stackrel{def}{=} \sum_{P \in \{D,T,N\}} \lambda_{E_P} Pr(E_P|Q); \quad (1)$$

where $\lambda_{E_D} + \lambda_{E_T} + \lambda_{E_N} = 1$. Next we describe how we compute these property scores in full details.

3.1 Entity Document Scoring

Scoring the entity document, in relation to the query, is based on the *profile based approach* for *EoS*, where entities are ranked according to the their profile similarity to the query. Specifically, to score the entity document, E_D, the graph we construct is composed of an entity document node and the query term nodes. (See the left graph in Figure 1). In analogy to [18] we define three clique types in such a graph. The first type is a two-node clique consisting of a query term q_i and the entity document node E_D. The feature function over T_{E_D}, the set of such cliques, measures how well each of the query terms represents the entity document. The function we use is based on a Dirichlet smoothed language model,

$$f_D^T(q_i, E_D) = \log\left[\frac{tf(q_i, E_D) + \mu \cdot cf(q_i)/|C|}{|E_D| + \mu}\right], \quad (2)$$

where $tf(x, Y)$ is the number of times x appears in Y, $cf(x)$ is the frequency of x in the corpus, $|C|$ is the total number of terms in the corpus, $|E_D|$ is the number of terms in the document E_D and μ is a smoothing parameter.

The second and the third click types are based on the dependence assumptions. O_{E_D} is the set of cliques for which the query terms in the clique appears contiguously in the query. The feature function over these cliques, $f_D^O(\#1(q_i...q_{i+k}), E_D)$, is defined by Equation 2, when replacing q_i with $\#1(q_i...q_{i+k})$, an ordered sub-sequence of $k+1$ query terms.

The third type of cliques, U_{E_D} contain an arbitrary subset of query terms $\{q_i...q_j\}$, together with the entity document node. Similarly to the above, the feature function over these cliques, $f_D^U(\#uwN(\{q_i...q_j\}), E_D)$, is defined by Equation 2, while replacing q_i with $\#uwN(\{q_i...q_j\})$, a text window of size N containing the given subset of query terms.

Finally, the entity document scoring function aggregates the feature functions over all clique types,

$$Pr(E_D|Q) \stackrel{def}{=} \sum_{I \in \{T,O,U\}} \lambda_{E_D}^I \sum_{c \in I_{E_D}} f_D^I(c) \qquad (3)$$

where $\lambda_{E_D}^T + \lambda_{E_D}^O + \lambda_{E_D}^U = 1$.

3.2 Entity Type Scoring

The second component of entity scoring is based on the correspondence of the entity type with the query target type. The graph G_T (See the middle graph in Figure 1) represents the joint probability distribution of two random variables - the entity type (E_T) and the target type (Q_T). Entities are scored based on the relationship between these two types: $Pr(E_T|Q) = f_T(c)$; f_T is the feature function defined over the single clique composed of the two nodes in G_T.

Entity types are determined based on the given data set and the entity retrieval task at hand. Therefore, the relationship between types should be defined specifically per corpus and task. In this work we use the INEX datasets, therefore, we measure the relationship strength between the entity type and the query type based on the similarity between the Wikipedia categories of the entity document and the query categories, as defined by the INEX topic.

The distance $d(E_T, Q_T)$ between the entity type E_T, and the query type Q_T, is calculated using the Wikipedia categories graph. Let E_{Ca} be the set of an entity categories and Q_{Ca} the set of categories defined by the INEX topic. If $Q_{Ca} \bigcap E_{Ca} \neq \phi$, we assume a full match and assign $d(Q_T, E_T)$ to zero. If not, the distance is defined to be the minimal path length between all pairs of categories of the two sets. When the distance of the entity category is far and exceeds a threshold on the maximum distance allowed, $d(Q_T, E_T)$ is set to this threshold. Additionally, when the entity category precedes all query categories in the graph the distance $d(Q_T, E_T)$ is set to that threshold.

Figure 2 shows a small twig of the Wikipedia category graph. Lets assume that the query categories are "novels" and "books"; if the entity category is "novels", then $d(Q_T, E_T) = 0$; if the entity category is "books by Paul Auster", then $d(Q_T, E_T) = 2$.

Based on the type distance between the entity and the query, the feature function is defined to be:

$$Pr(E_T|Q) \stackrel{def}{=} f_T(c) = log \left[\frac{e^{-\alpha d(Q_T, E_T)}}{\sum_{E' \in R} e^{-\alpha d(Q_T, E'_T)}} \right]; \qquad (4)$$

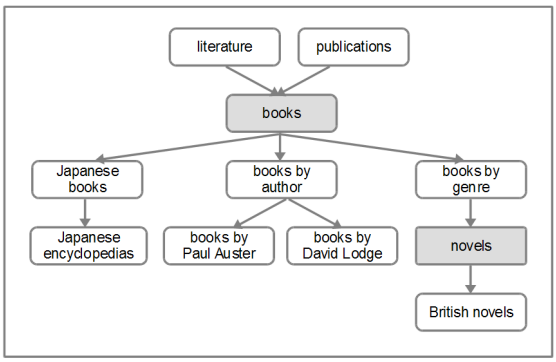

Figure 2: A Twig of the Wikipedia category graph

α is a decay coefficient and R is a list of entity documents retrieved by the entity document based retrieval. (See Equation 3.) For integration with the other entity scores, we normalize the type-based score (as well as the document-based score) by the sum of type-based scores of all entities in the list being ranked.

3.3 Entity Name Scoring

An entity can have different names and several entities can have the same name. For simplicity, we do not address name resolution and disambiguation in this work. As our study focuses on Wikipedia datasets, we use the title of the entity document as the unique entity name.

The graph G_N (right graph in Figure 1) models the joint distribution of the entity name and the query terms. To score the entity name in relation to the query, we define a set of cliques in G_N, as well as feature functions over those cliques. We use four clique types for entity name scoring. The first one, S_{E_N}, is a one-node clique containing the entity name node alone. Additionally, in analogy to the cliques defined for entity document scoring, we define three clique types; T_{E_N}, the set of two-node cliques containing the entity node name together with a single query term; O_{E_N}, a multi-node clique containing two or more contiguous query terms, and U_{E_N} which contains two or more arbitrary query terms.

We use two types of feature functions over these cliques.

Voting approach. The first type, inspired by The *voting approach*, uses a list of scored document entities, retrieved by the entity document scoring model (using Equation 3). According to this approach, an entity is considered relevant if many top-scored documents "vote" for it, i.e., it is referred to by many of those "relevant" entities. For a single, one entity node clique, the voting-based feature function is computed by:

$$f_N^S(c) = log\,[Pr(E_N)] = \log \left[\sum_{E'_D \in R} Pr(E_N|E'_D) Pr(E'_D) \right];$$

R is the list of entity documents most highest ranked. $Pr(E'_D)$ is the prior probability of document E'_D being relevant, which is assumed to be uniform over all entity documents. $Pr(E_N|E'_D)$ is the generation probability assigned to the entity name by a Dirichlet-smoothed language model induced from E'_D; here and after, we refer to probabilities assigned by language models as generation probabilities.

The feature function over two-node cliques, T_{E_N}, estimates the joint probability distribution of the entity name with one of the query terms, q_i, by the weighted sum of the entity name and query term generation probabilities over the list of documents:

$$f_N^T(c) = \log\left[Pr(E_N, q_i)\right] = \log\left[\sum_{E'_D \in R} Pr(E_N|E'_D, q_i) Pr(E'_D, q_i))\right] = \log\left[\sum_{E'_D \in R} Pr(E_N|E'_D, q_i) Pr(q_i|E'_D) Pr(E'_D)\right].$$

$Pr(E_N|E'_D, q_i)$, the generation probability of the entity name, is estimated by

$$Pr(E_N|E'_D, q_i) = \frac{tf(\#uwN(E_N, q_i), E'_D) + \mu \cdot cf(\#uwN(E_N, q_i))/|C|}{|E'_D| + \mu},$$

where $tf(\#uwN(E_N, q_i), E'_D)$ is the number of times the terms of the entity name E_N and the query term q_i appear in any order within a window of size N in the document E'_D, and $cf(\#uwN(E_N, q_i))$ is the total count of this expression in the whole corpus. $Pr(q_i|E'_D)$ is the probability for generating query term q_i from the E'_D language model.

Similarly, for the multi-node cliques, we define the feature functions, $f_N^O(c)$ and $f_N^U(c)$; the first is defined over contiguous query term cliques; the second over the arbitrary query term cliques. (Refer back to section 3.1.) These feature functions are defined in a similar manner to $f_N^T(c)$, replacing $\#uwN(E_N, q_i)$ with $\#uwN(E_N, \#1(q_i...q_{i+k}))$ and $\#uwN(E_N, q_i...q_j)$, respectively.

Global approach. The estimate above is local in the sense that it is computed based on the highest ranked documents retrieved in response to the query. Next we define a feature functions over the graph cliques which uses a "global measure", computed based on a whole corpus information, to estimate the joint probability of the entity name and the query terms. The measure calculated for each of the two-node cliques is in the spirit of a pointwise mutual information (PMI) over the corpus, which estimates the semantic relationship strength between the clique terms:

$$PMI(E_N, q_i) \stackrel{def}{=} \log\left[\frac{cf(E_N) \cdot cf(q_i)}{cf(\#uwN(E_N, q_i))}\right],$$

where $cf(E_N)$, $cf(q_i)$ and $cf(\#uwN(E_N, q_i))$ are the total counts of the entity name, the query term, and both (within a window of size N) in the corpus, respectively.

The feature function over the two-node cliques $f_N^{PMI_T}$, is defined to be the PMI score normalized by the sum of PMI scores of all entities being ranked. Similarly, for the multi-node cliques, the feature functions $f_N^{PMI_O}$ and $f_N^{PMI_U}$ are defined; the first is defined over contiguous query term cliques and the second over arbitrary term cliques. We apply the same computation as for $f_N^{PMI_T}$, while replacing q_i with $\#1(q_i...q_{i+k})$ and $\#uwN(q_i...q_j)$, respectively.

The final entity name score aggregates all feature function values over all cliques:

$$Pr(E_N|Q) \stackrel{def}{=} \sum_{X \in A} \lambda_{E_N}^X \cdot \sum_{c \in X_{E_N}} f_N^X(c) \qquad (5)$$

where $A = \{S, T, O, U, PMI_T, PMI_O, PMI_U\}$, $\sum_{X \in A} \lambda_{E_N}^X = 1$.

Data set	WP year	Collection size	#Documents in collection	Train topics	Test topics
2007	2006	4.4 GB	659,388	28	46
2008				74	35
2009	2008	50.7 GB	2,666,190	-	55

Table 1: INEX entity ranking datasets

3.4 Final Entity Scoring

By Equation 1, the final score of an entity is computed by a linear aggregation of the three entity properties scores. The scoring process is performed in four stages. At first, an initial list of ranked entity documents is retrieved using Equation 3. Second, the entities in the list are re-ranked based on the aggregation of the entity document and entity type scores. In the third stage, additional entities are retrieved by collecting referred entities (e.g., Wikipedia outlinks) from the top retrieved document entities in the list. To control the amount of additional entities to score, only those with a relatively high type score are added to the list (Specifically, their category-based distance from the query is 1 or less). Finally, all entities are re-ranked based on the final score assigned by Equation 1.

4. EVALUATION

4.1 Experimental Setup

To evaluate our proposed model we conducted a set of experiments using the datasets of INEX *entity ranking track* of 2007, 2008, and 2009. We use these testbeds for several reasons. First, entity extraction over the data is not required since the entities are defined to be Wikipedia pages. This enables us to focus on the retrieval task itself and not on related issues such as entity extraction technologies. Second, constructing entity documents in not required since an entity is defined as an item having a Wikipedia page. We treat this page as the entity document. Third, the entity type can be directly inferred from the entity page's categories. Finally, the topics of the *entity ranking task* are well defined and can be utilized by our model which looks for relevant entities to an ad-hock topic.

Data. The INEX *entity ranking tracks* in 2007 and 2008 use the English Wikipedia dataset from 2006, composed of 659,388 documents (4.4GB). The 2009 track uses the English Wikipedia from 2008, composed of 2,666,190 documents (50.7GB).

The topics for 2007 were divided to train and test topics. 28 Previous INEX ad hoc topics were adapted to the entity ranking task and used for training. The test topics consisted of 46 topics. In 2008, the 2007 topics were used for training while 35 new topics were added for testing. The topics used for testing in 2009 are 55 topics out of 60 test topics used in 2007 and 2008, while no topics were devoted for training. Table 1 provides a summary of the collections and topics used.

Following the INEX guidelines, the metric used for evaluation for INEX 2007 is the mean average precision (MAP). For the INEX 2008 and 2009 the *infAP* metric was used [22].

Entity Property	Symbol	Parameter name	Value
E_D	N	query proximity window size	10
E_T	d_{max}	max. distance in categories graph	5
	α	category score decay coefficient	3
E_N	R	# docs for computing voting score	500
	K	entity terms proximity	3
	N	entity and query terms proximity	10
	R_{init}	# docs for entity expansion	50

Table 2: Parameters of the model which were set to specific selected values.

Implementation details. The two Wikipedia collections were indexed using Apache Lucene[1]. The Wikipedia pages were tokenized using Lucene snowball analyzer which performs stop words removal and Porter stemming. We implemented an MRF based retrieval method using the built-in proximity search operator (*SpanNearQuery*) of Lucene.

Parameters tuning. We divide the parameters composing the entities ranking formula into two types. Parameters of the first type, given in Table 2, include those which are specific for each entity property ranking score. The values of these parameters were selected based on their effect on the model performance (more details to follow) after extensive search over a wide a range of values. The selected values were used for all experiments.

The second type of parameters includes the relative weights of the different entities properties, the feature functions defined for each property graph, and the smoothing parameters of the language model based feature function. Parameters of this type were optimized over the training topics using the **coordinate ascent (CA)** algorithm proposed by Metzler and Croft for MRF model tuning [17].

The tuning process was performed in stages, following the scoring process described in section 3.4. At first, the smoothing parameters of the Language Model scoring function, as well as the relative weights of the entity document scoring function (see Equation 3) were tuned using CA. The target metric to be optimized was the MAP or infMAP of the retrieved entities lists. The obtained weights and smoothing parameters were then fixed for later stages.

At the second stage, the relative weights of the entity document score and entity type score (λ_{E_D} and λ_{E_T}) were tuned and fixed using CA. By using the selected values, an initial list of re-ranked entities was created, and additional entities were retrieved, as described in Section 3.4.

At the final stage, the relative entity properties weights, as well as the weights of the entity name scoring function were tuned using CA. For simplicity, to reduce the number of free parameters, we set $\widetilde{\lambda}^X_{E_N} = \lambda^X_{E_N} \cdot \lambda_{E_N}$, $(X \in \{S,T,O,U\})$, and $\widetilde{\lambda}^{PMI_Y}_{E_N} = \lambda^{PMI_Y}_{E_N} \cdot \lambda_{E_N}$ where $(Y \in \{T,O,U\})$.

The optimization process was performed separately for each dataset. For the 2007 and 2008 datasets the optimal parameters were found using the train topics and then the model performance was estimated using the test topics. For the 2009 dataset no training topics were available so a 5-fold cross-validation over the test topics was performed for parameter tuning.

[1]http://lucene.apache.org/core/

Entity Score	Parameter symbol	2007	2008	2009
$S(E_D)$	μ	100	100	800
	$\lambda^T_{E_D}$	0.83	0.87	0.9
	$\lambda^O_{E_D}$	0.05	0.05	0.04
	$\lambda^U_{E_D}$	0.12	0.08	0.06
$S(E_D, E_T)$	λ_{E_D}	0.42	0.45	0.7
	λ_{E_T}	0.58	0.55	0.3
$S(E_D, E_T, E_N)$	λ_{E_D}	0.19	0.19	0.42
	λ_{E_T}	0.42	0.42	0.35
	$\widetilde{\lambda}^S_{E_N}$	0.01	0.04	0
	$\widetilde{\lambda}^T_{E_N}$	0.05	0.02	0.01
	$\widetilde{\lambda}^O_{E_N} = \widetilde{\lambda}^U_{E_N}$	0.005	0.01	0
	$\widetilde{\lambda}^{PMI_Y}_{E_N}, (Y \in \{T,O,U\})$	0.11	0.11	0.07

Table 4: Parameters of the model which were tuned using the CA algorithm.

The entities scores computed at the different stages are denoted as follows: $S(E_D)$ is the entity score composed of the entity document score alone (see Equation 3); $S(E_D, E_T)$ is the entity score composed of the document and the type score; and, $S(E_D, E_T, E_N)$ is the full entity score (see Equation 1).

4.2 Experimental Results

Initial Retrieval ($S(E_D)$). Table 3 shows the performance obtained while using the entity document score computed under various independence assumptions (rows 1, 4 and 7). Surprisingly, there is no significant difference between the various dependence models, i.e., there is no significant advantage in using the dependence models over the independence model.

Considering the entity type ($S(E_D, E_T)$). The performance of aggregating the entity document and the entity type scores is presented in Table 3 (rows 2, 5 and 8). Adding the entity type score to the entity document score yields a statistically significant improvement. This holds for all three data sets as well as for each of the dependence assumptions. However, a comparison of the performance obtained when using different dependence assumptions shows that as before, no statistically significant difference is found for the three datasets.

Final scoring ($S(E_D, E_T, E_N)$). Table 3 shows the performance of using the full score that is composed of the entity document score, the entity type score, and the entity name score (rows 3, 6 and 9). Using the entities full score yields performance that is statistically significantly better than that of using the document score alone and the document and type score. However, as before, no statistical significance difference is found when comparing the full scores obtained using the different dependence assumptions.

Comparing the performance of the full model for each dataset (MAP or infMAP), to the best performance attained by the INEX track participants, shows that our model consistently outperforms the official track results in 2007 and 2008, and posts performance that is similar to the best reported for 2009. We note that the best performance obtained in 2009 is 0.517 [4]. This performance is far better

Dependence assumption	Row Number	Score type	2007 (MAP)	2008 (infMAP)	2009 (infMAP)
Full Independence (FI)	1	$S(E_D)$	0.202	0.126	0.19
	2	$S(E_D, E_T)$	0.305* (+50.99%)	0.282* (+123.81%)	0.240 (+26.32%)
	3	$S(E_D, E_T, E_N)$	0.333* (+64.85%)	0.359* (+184.92%)	0.245* (+28.95%)
Sequential Dependence (SD)	4	$S(E_D)$	0.205	0.137	0.201
	5	$S(E_D, E_T)$	0.316* (+54.15%)	0.280* (+122.22%)	0.252 (+25.37%)
	6	$S(E_D, E_T, E_N)$	0.338* (+64.88%)	0.364* **(+165.69%)**	0.258* **(+28.36%)**
Full Dependence (FD)	7	$S(E_D)$	0.205	0.133	0.198
	8	$S(E_D, E_T)$	0.308* (+50.24%)	0.278* (+109.02%)	0.25 (+26.26%)
	9	$S(E_D, E_T, E_N)$	0.340* **(+65.85%)**	0.353* (+165.41%)	0.256 (+29.29%)
INEX top			0.306	0.341	0.27

Table 3: Comparison of different independence assumptions and different entity score types. Values in parenthesis denote the relative improvements over using entity document score ($S(E_D)$). $'*'$ indicates a statistically significant difference with using only the entity document score.

than that of all other methods, but it was obtained using relevance feedback. For a fair comparison, we compare our model to the best performing method that did not use feedback.

Model parameters. The values of the various parameters of the sequential dependence model, as set by the CA algorithm, are shown in Table 4. As can be seen, the optimal parameters values for the datasets of 2007 and 2008 are similar; yet, these are different than those for the 2009 data set. This could be explained by the fact that the same Wikipedia collection is used in 2007 and 2008 while a different, new collection was used in 2009. Examination of the full entity score weights shows that the entity document score, the entity type score and the global entity name score are all assigned with a substantial weight. The features of the voting-based score were assigned with a relatively very low weight, compared to the global (PMI) based features.

5. SUMMARY

We presented an entity ranking model which integrates the profile approach, the voting approach, and the filtering approach, in a unified way using the MRF framework. Experiments performed with our model over the INEX *entity ranking track* datasets showed that it performs substantially better than the leading INEX systems in 2007 and 2008, and similarly to the best performing systems in 2009. Using various dependence assumptions did not result in significant improvement in the model performance over using the full independence assumption. For future work we intend to explore this model with additional data collections, specifically, a web collection. Using additional entity properties is an interesting direction for further research as well.

Acknowledgments We thank the anonymous reviewers for their comments. This paper is based upon work supported in part by the Israel Science Foundation under grant no. 557/09, by IBM's SUR award, and by Google's and Yahoo!'s faculty research awards. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

6. REFERENCES

[1] L. Azzopardi, K. Balog, and M. Rijke. Language modeling approaches for enterprise tasks. In *TREC*, 2006.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of SIGIR*, pages 43–50, 2006.

[3] K. Balog, M. Bron, and M. De Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29(4), 2011.

[4] K. Balog, M. Bron, M. de Rijke, and W. Weerkamp. Combining term-based and category-based representations for entity search. In *Focused Retrieval and Evaluation*, volume 6203, pages 265–272. 2010.

[5] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *TREC*, 2009.

[6] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2010 entity track. In *TREC*, 2010.

[7] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2011 entity track. In *TREC*, 2011.

[8] L. Bonnefoy and P. Bellot. Lia-ismart at the TREC 2011 entity track : Entity list completion using contextual unsupervised scores for candidate entities ranking. In *TREC*, 2011.

[9] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 enterprise track. In *TREC*, 2005.

[10] A. de Vries, A.-M. Vercoustre, J. Thom, N. Craswell, and M. Lalmas. Overview of the INEX 2007 entity ranking track. In *Focused Access to XML Documents*, volume 4862, pages 245–251. 2008.

[11] G. Demartini, A. de Vries, T. Iofciu, and J. Zhu. Overview of the INEX 2008 entity ranking track. In *Advances in Focused Retrieval*, volume 5631, pages 243–252. 2009.

[12] G. Demartini, C. Firan, and T. Iofciu. L3s at inex 2007: Query expansion for entity ranking using a highly accurate ontology. In *Focused Access to XML Documents*, volume 4862, pages 252–263. 2008.

[13] G. Demartini, T. Iofciu, and A. de Vries. Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation*, volume 6203, pages 254–264. 2010.

[14] S. Fissaha Adafre, M. de Rijke, and E. Tjong Kim Sang. Entity Retrieval. In *Recent Advances in Natural Language Processing (RANLP 2007)*, 2007.

[15] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma. Thuir at TREC 2005: Enterprise track. In *TREC*, 2005.

[16] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of SIGIR*, pages 267–274, 2009.

[17] D. Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10:257–274, 2007.

[18] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of SIGIR*, pages 472–479, 2005.

[19] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *Proceedings of CIKM*, pages 731–740, 2007.

[20] Z. Ru, Y. Chen, W. Xu, and J. Guo. TREC 2005 enterprise track experiments at bupt. In *TREC*, 2005.

[21] O. Vechtomova. Related entity finding: University of waterloo at TREC 2010 entity track. In *TREC*, 2010.

[22] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of SIGIR*, pages 603–610, 2008.