Ranking Related Entities: Components and Analyses

Marc Bron
m.m.bron@uva.nl

Krisztian Balog
k.balog@uva.nl

Maarten de Rijke
derijke@uva.nl

ISLA, University of Amsterdam
Science Park 904, 1098 XH Amsterdam

ABSTRACT

Related entity finding is the task of returning a ranked list of homepages of relevant entities of a specified type that need to engage in a given relationship with a given source entity. We propose a framework for addressing this task and perform a detailed analysis of four core components; co-occurrence models, type filtering, context modeling and homepage finding. Our initial focus is on recall. We analyze the performance of a model that only uses co-occurrence statistics. While it identifies a set of related entities, it fails to rank them effectively. Two types of error emerge: (1) entities of the wrong type pollute the ranking and (2) while somehow associated to the source entity, some retrieved entities do not engage in the right relation with it. To address (1), we add type filtering based on category information available in Wikipedia. To correct for (2), we add contextual information, represented as language models derived from documents in which source and target entities co-occur. To complete the pipeline, we find homepages of top ranked entities by combining a language modeling approach with heuristics based on Wikipedia's external links. Our method achieves very high recall scores on the end-to-end task, providing a solid starting point for expanding our focus to improve precision; additional heuristics lead to state-of-the-art performance.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Entity search, Language modeling, Wikipedia

1. INTRODUCTION

Over the past decade, increasing attention has been devoted to retrieval technology aimed at identifying entities relevant to an information need. The area received a big boost with the arrival of the TREC Question Answering track in 1999, where much research has focused on fact-based questions such as "Who invented the paper clip?" Such questions can be answered by named entities such as locations, dates, etc. [35]. The expert finding task, studied at the TREC Enterprise track (2005–2008), focused on a single type of entity: people [7]. The INEX Entity Ranking task (2007–2009) broadened the task to include other types and required systems to return ranked lists of entities given a textual description ("Countries where one can pay with the euro") and type information ("countries") [9]. Next, the TREC 2009 Entity track introduced the *related entity finding* (REF) task: given a source entity, a relation and a target type, identify homepages of target entities that enjoy the specified relation with the source entity and that satisfy the target type constraint [3]. E.g., for a source entity ("Michael Schumacher"), a relation ("His teammates while he was racing in Formula 1") and a target type ("people") return entities such as "Eddie Irvine" and "Felipe Massa." REF aims at making arbitrary relations between entities searchable; it provides a way of searching for information through entities, previously only possible by (implicitly) manually annotated links such as those in social networks.

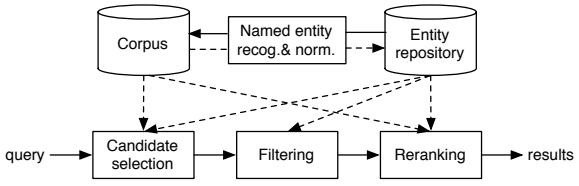


Figure 1: Components of an idealized entity finding system. Solid arrows indicate control flow, dashed arrows data flow.

We start with an idealized entity retrieval architecture, see Fig. 1. Computations take place at two levels: the entity repository is built off-line, using tools and techniques for named entity recognition and normalization. Queries are processed online, through a retrieval pipeline. This pipeline resembles a question answering architecture, where first candidate answers are generated, followed by type filtering and the final ranking (scoring) steps. Candidate generation is a recall-oriented step, while the subsequent two blocks aim to improve precision. Our work sets out the challenge of adopting this general architecture to the REF task, and addresses the issue of balancing precision and recall when executing a search.

When building a system to perform a task such as REF, the most important evaluation is on the end-to-end task. The TREC Entity track will play an important role in advancing REF technology, but its end-to-end focus means that it is difficult to disentangle the performance contributions of individual components. This effect is reinforced in the case of a new task such as REF where a canonical architecture has yet to emerge. In this paper we go through a series of ablation studies and contrastive runs so as to obtain an under-

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'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

standing of each of the components that play a role and the impact they have on precision and recall.

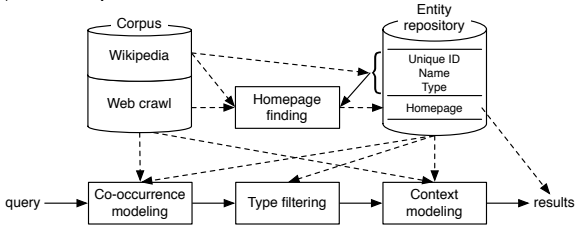


Figure 2: Components of our REF system.

Specifically, we address the REF task as defined at TREC 2009 and consider a particular instantiation of the idealized entity finding system, shown in Fig. 2. Our focus is on retrieval and ranking rather than on named entity recognition and normalization; to simplify matters we use Wikipedia as a repository of (normalized) known entities. While the restriction to entities in Wikipedia is a limitation in terms of the number of entities considered, it provides us with high-quality data, including names, unique identifiers and type information, for millions of entities. Our framework is generic and conceptually independent of this particular usage of Wikipedia.

Given our focus on entities in Wikipedia, it is natural to address the REF task in two phases. In the first we build up our retrieval pipeline (the blocks at the bottom of Fig. 2) working only with Wikipedia pages of entities; in the second we map entities to their homepages. In phase one we use a generative framework to combine the components. The first component is a co-occurrence-based model that selects candidate entities. While, by itself, a co-occurrence-based model can be effective in identifying the potential set of related entities, it fails to rank them effectively. Our failure analysis reveals two types of error that affect precision: (1) entities of the wrong type pollute the ranking and (2) entities are retrieved that are associated with the source entity without engaging in the right relation with it. To address (1), we add a type filtering component based on category information in Wikipedia. To correct for (2), we complement the pipeline with contextual information, represented as statistical language models derived from documents in which the source and target entities co-occur. The addition of context proves beneficial for both recall and precision. A final improvement in this phase is obtained by employing a large corpus to correct for sparseness issues.

In phase two, we conform to the official TREC definition of the REF task by adding a homepage finding component that maps entities represented by Wikipedia pages to homepages. We show that our approach achieves competitive performance on the official task, especially in terms of recall. We demonstrate the generalizability of our framework by expanding it with two heuristics: one aimed at improving type filtering, the other at co-occurrence. We find that these methods have a very positive impact on all measures.

The main contribution of this paper is two-fold. First, we propose a transparent architecture for addressing the REF task. Second, we provide a detailed analysis of the effectiveness of its components and estimation methods, shedding light on the balance between precision and recall in the context of the REF task.

Below, we discuss related work in §2. In §3 we detail our approach to the REF task. Our experimental setup is described in §4. In §5 we analyze the effectiveness of a pure co-occurrence model. Type filtering is considered in §6 and contextual information is added in §7. Improved estimations of co-occurrence and context models are considered in §8. We address the (sub)task of homepage finding (mapping entities to their homepages) in §9. In §10 we discuss TREC Entity results as well as additional heuristics that can be incorporated into our framework. We conclude in §11.

2. RELATED WORK

Entity retrieval. The roots of entity retrieval go back to natural language processing, specifically to information extraction (IE). The goal in IE is to find all entities for a certain class, for example "cities." The general approach taken uses context based patterns to extract entities; e.g., "cities such as * and *", either learned from examples [28] or created manually [16]. At the intersection of natural language processing and IR lies question answering (QA), which combines IE and IR, investigated at the TREC QA track [35]. What sets QA apart from Entity Retrieval? One is a matter of technology: many QA systems considered at TREC have a knowledge-intensive pipeline that simply does not comply with the wishes of efficient processing on very large volumes of data. Another is the difference in task; while the "list" subtask at the QA track does indeed resemble the REF task, the two differ in important ways: (i) QA list queries do not always contain an entity [34], e.g., "What are 12 types of clams" and (ii) REF queries impose a more specific (elaborate) relation between the source entity and the target entities, e.g., "Airlines that currently use Boeing 747 planes."

A particularly relevant paper on the interface of QA and REF is [26] wherein entity language models are processed using a probabilistic representation of the language used to describe a named entity (person, organization or location). The model is constructed from snippets of text surrounding mentions of an entity. Unsurprisingly, we model the language model of an individual entity in the same manner, but have more complex models of pairs of entities.

Our main concern is with precision and recall aspects of our approach to the REF task. We initially focus on recall and then apply techniques to boost precision: one of these techniques is *type filtering*, aimed at demoting entities that are not of the required type. Previously, *type filtering* has been considered in the setting of QA, where candidate answers are filtered by employing surface patterns [27] or by a mix of top-down and bottom-up approaches [29]. We apply type filtering based on Wikipedia category assignments and category structure in the context of the REF task.

The expert finding task, which was run at the TREC Enterprise track [7], focuses on a single type ("person") and relation ("expert in"). In a language modeling approach to the task experts are found either by modeling an expert's knowledge by its associated documents ("Model 1") or first collecting topic related documents and then modeling experts ("Model 2") [1, 2]. Additionally, kernels have been used to emphasize terms occurring in close proximity to experts (entities) [25]. A two stage language modeling approach, consisting of a relevance and a co-occurrence model, has been considered in [4]; the relevance model determines if a document is relevant to a query, while the co-occurrence model determines the association between an expert and query in a document. A generative probabilistic framework was proposed in [11], with two families of approaches: candidate and topic generation models.

The novelty of our approach is that we use co-occurrence and context to model entity-entity associations, instead of entity-document and document-query associations as seen in most expert finding systems. Co-occurrence-based methods are widely used to determine the strength of association between terms. These methods come in many flavors, e.g., as global co-occurrence [18] (determined on an entire corpus) or local co-occurrence [41] (determined on a relevant set of documents) for query expansion. Hasegawa et al. [15] use the context of entity co-occurrence pairs for relation extraction; entity pairs that occur in the same sentence are clustered based on terms between them, and relations are characterized by frequent words in the cluster. Maximum likelihood estimation (MLE) is an obvious choice for determining co-occurrence; a problem is that some words co-occur frequently just by chance. In

[21] a number of hypothesis testing methods are listed that determine whether the co-occurrence of two entities is more than mere chance, these include statistical tests, likelihood ratio and point wise mutual information. In §5 we determine their value for REF.

The INEX Entity Ranking track [9] broadened expert search by moving from searching for a single type of entity (people) to any type using Wikipedia categories. The corpus also changed to Wikipedia, so that each entity corresponds to a Wikipedia page. Two tasks were introduced, entity ranking: return a list of Wikipedia pages (entities) for a query and category, and list completion: return a list of Wikipedia pages for a query, category and example entities. The fact that each entity is represented by a Wikipedia page allows for using standard document retrieval to obtain a list of relevant entities for a query. The approaches differ in the way they combine this with category and example entity information; using a language modeling framework to combine initial retrieval scores with category information [17, 39], or a linear combination of document, category and link based component scores [8, 33, 45]. To derive a score for the category component most approaches use set overlap between entity categories and topic target categories; others use the topic category label as a query to an index of category labels [8, 45]. Another commonly used technique is category expansion, based either on the Wikipedia category structure [32, 39] or on lexical similarity between category labels and the query topic [33].

TREC 2009 Entity track. A recent development in evaluating entity-oriented search was the introduction of the Entity track at TREC in 2009 with the aim to perform entity-oriented search tasks on the web [3]. The first edition featured the related entity finding (REF) task. One approach to this task is to directly obtain homepages by submitting the REF query (source entity and relation) to a search engine [24]. Other approaches first collect text snippets from documents relevant to the REF query, next obtain entities by performing named entity recognition on the snippets, implement some sort of ranking step and finally find homepages, usually by using entity names as queries [37]. Several language modeling approaches were employed to rank entities, where the entity model is constructed from snippets containing the entity and the relation is used as a query [40, 42, 44]. The two stage retrieval model from the Enterprise track is adapted in [38]. Fang et al. [12] use a hierarchical relevance retrieval model and improve their model by exploiting list structures, training regression models for type filtering and applying heuristic filtering and pattern matching rules. Zhai et al. [43] propose a probabilistic framework to estimate the probability of an entity given a REF query, with two components: the probability of the relation given an entity and source entity, and the probability of an entity given the source entity and target type. While this model is the closest to our approach, it differs in the assumptions made about the dependencies between the query components, see §3. The approaches further differ in the way they estimate co-occurrence and construct entity and relation models. A number of approaches rely heavily on Wikipedia; as a repository of entity names, to perform entity type filtering based on categories and to find homepages through external links [19, 22, 30].

3. APPROACH

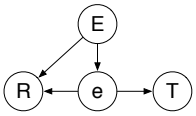
The goal of the REF task is to return a ranked list of relevant entities e for a query, where a query consists of a source entity (E), target type (T) and a relation (R) [3]. We formalize REF as the task of estimating the probability $P(e|E,T,R)$. This probability is difficult to estimate, due to the lack of training material, exacerbated by the fact that relations do not come from a closed vocabulary. Also, the model should capture a particular relation conditioned on the two entities involved. To address these concerns we turn to a generative model. First, we apply Bayes' Theorem and rewrite $P(e|E,T,R)$ to:

$$P(e|E,T,R) = \frac{P(E,T,R|e)\cdot P(e)}{P(E,T,R)}. \quad (1)$$

Next, we drop the denominator as it does not influence the ranking of entities, and derive our final ranking formula as follows:

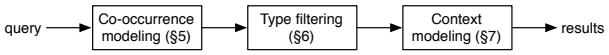
$$\begin{aligned} &P(E,T,R|e)\cdot P(e) \\ &\quad \propto P(E,R|e)\cdot P(T|e)\cdot P(e) \quad (2)\\ &\quad = P(E,R,e)\cdot P(T|e) = P(R|E,e)\cdot P(E,e)\cdot P(T|e) \\ &\quad = P(R|E,e)\cdot P(e|E)\cdot P(E)\cdot P(T|e) \quad (3)\\ &\quad \stackrel{\text{rank}}{=} P(R|E,e)\cdot P(e|E)\cdot P(T|e) \quad (4)\end{aligned}$$

In (2) we assume that type T is independent of source entity E and relation R. We rewrite $P(E,R|e)$ to $P(R|E,e)$ so that it expresses the probability that R is generated by two (co-occurring) entities (e and E). Finally, we rewrite $P(E,e)$ to $P(e|E)\cdot P(E)$ in (3) as the latter is more convenient for estimation. We drop $P(E)$ in (4) as it is assumed to be uniform, thus does not influence the ranking.



The generative model, shown above, functions as follows. The input entity E is chosen with probability $P(E)$, which generates a target entity e with probability $P(e|E)$. The input and target entities together generate a relation R with probability $P(R|E,e)$. Finally, the target entity generates a type T with probability $P(T|e)$.

Assuming that input entities are chosen from a uniform distribution, we are left with the following components: (i) pure co-occurrence model ($P(e|E)$), (ii) type filtering ($P(T|e)$) and (iii) contextual information ($P(R|E,e)$). We summarize the developments to come in the figure below; we analyze a pure co-occurrence model and its performance on the REF task in §5. We then add type filtering and contextual information to the pipeline; these are introduced and examined in §6 and §7, respectively. The components are combined using Eq. 4.



4. EXPERIMENTAL SETUP

Research questions. We address the official REF task, REF on a web corpus, by first solving it on a smaller less noisy corpus: Wikipedia, where entities are identified by their Wikipedia page. In this setting we consider three research questions. (1) How do different measures for computing co-occurrence affect the recall of a pure co-occurrence based REF model? (2) Can a basic category based type filtering approach successfully be applied to REF to improve precision without hurting recall? (3) Can recall and precision be enhanced by adding context to co-occurrences, to ensure that source and target entities engage in the right relation? We then look at the REF task in the setting of a large web corpus. To conform to the official REF task we map the Wikipedia entity representation to a representation that identifies entities by homepages and consider three additional research questions. (4) Does the use of a larger corpus improve estimations of co-occurrence and context models? (5) Is the initial focus on Wikipedia a sensible approach; can it achieve comparable performance to other approaches? (6) Can our basic

ID	Source entity (E)	Relation (R)	Type (T)
1	Blackberry	Carriers that Blackberry makes phones for.	ORG
4	Philadelphia, PA	Professional sports teams in Philadelphia.	ORG
5	Medimmune, Inc.	Products of Medimmune, Inc.	PROD
6	Nobel Prize	Organizations that award Nobel prizes.	ORG
7	Boeing 747	Airlines that currently use Boeing 747 planes.	ORG
9	The Beaux Arts Trio	Members of The Beaux Arts Trio.	PER
10	Indiana University	Campuses of Indiana University.	ORG
11	Home Depot Foundation	Donors to the Home Depot Foundation.	ORG
12	Air Canada	Airlines that Air Canada has code share flights with.	ORG
14	Bouchercon 2007	Authors awarded an Anthony Award at Bouchercon in 2007.	PER
15	SEC conference	Universities that are members of the SEC conference for football.	ORG
17	The Food Network	Chefs with a show on the Food Network.	PER
18	Jefferson Airplane	Members of the band Jefferson Airplane.	PER
19	John L. Hennessy	Companies that John Hennessy serves on the board of.	ORG
20	Isle of Islay	Scotch whisky distilleries on the island of Islay.	ORG

Table 1: Description of our 15 test topics. Target entity types are ORG=organization, PER=person and PROD=product.

framework effectively incorporate additional heuristics in order to be competitive with other state-of-the-art approaches?

Document collection. Our document collection is the ClueWeb09 Category B subset [5] ("CW-B" for short), with about 50 million documents, including English Wikipedia. We use the Wikipedia part of ClueWeb09 and filtered out duplicate pages, page not found errors and non-English pages. This left us with about 5M documents, 2.6M of which correspond to unique entities. The total number of unique entity occurrences in Wikipedia documents (i.e., each unique entity occurring in a document counts only once, independent of the actual number of occurrences) is 373M.

Entity recognition and normalization. While named entity recognition and normalization are not our focus, they are key pre-processing steps. We use Wikipedia as a repository of known (normalized) entities. We handle named entity recognition (NER) in Wikipedia by considering only anchor texts as entity occurrences. We obtain an entity's name by removing the Wikipedia prefix from the anchor URL. For named entity normalization (NEN) we map URLs to a single entity variant. Here we make use of Wikipedia redirects that map common alternative spellings or references (e.g., "Schumacher," "Schumi" and "M. Schumacher") to the main variant of an entity ("Michael Schumacher"). Below, when using the full CW-B subset, we use the entity names as queries to an index of this collection. This effectively bypasses NER as the resulting document lists identify in which documents the entities occur. In this case, we do not perform NEN; while potentially introducing noise, we believe that the amount of data partly compensates for this.

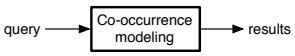
Test topics. We base our test set on the TREC 2009 Entity topics. A topic consists of a source entity (E), a target entity type (T) and the desired relation (R) described in free text. Since we are restricting ourselves to entities in Wikipedia, we are not able to use all 20 TREC Entity topics, but only 15 of them. Specifically, we exclude three topics (#3, #8 and #13) without relevant results in Wikipedia and another two (#2 and #16) with source entities without a Wikipedia page. For the remaining topics we manually mapped the source entity to a Wikipedia page, this is the only manual intervention in the pipeline; the topics are listed in Table 1.

We perform two types of evaluation. First, throughout §5–§8 we focus on finding entities as represented by their Wikipedia page. We establish ground truth by extracting all primary Wikipedia pages from the TREC 2009 Entity qrels. We handle Wikipedia redirects and duplicates in our evaluation; a Wikipedia page returned for any of the variants of a relevant entity is considered to be correct, but once found, other variants of that page are ignored. This setup constitutes a change to the original TREC REF task, arguably making it easier, therefore the reported numbers are not directly comparable with those of the TREC 2009 Entity track [3]. Our second type of evaluation, on the original TREC REF task, is performed in §10, where we compare our scores with those of TREC Entity participants; based on their original submissions, we also compute their Wikipedia-based evaluation scores.[1]

Evaluation metrics. We focus on two measures: precision and recall. Specifically, we use R-Precision (R-prec), where R is the number of relevant entities for a topic, and recall at rank N (R@N), where we take N to be 100, 2000 and "All" (i.e., considering all returned entities). In Table 10 we also report on the metrics used at the TREC 2009 Entity track: P@10, NDCG@R, and the number of primary and relevant entity homepages retrieved. We forego significance testing as we do not have the minimal number of topics (25) recommended [36].

5. CO-OCCURRENCE MODELING

The pure co-occurrence component is the first building block of our retrieval pipeline. It can produce a ranking of entities on its own.



Since we are planning on expanding this pipeline with additional components (that will build on the set of entities identified in this step), our main focus throughout this section will be on recall.

Estimation. The pure co-occurrence component ($P(e|E)$) expresses the association between entities based on occurrences in documents, independent of context (i.e., the actual content of documents). To express the strength of co-occurrence between e and E we use a function $\text{cooc}(e, E)$ and estimate $P(e|E)$ as follows:

$$P(e|E) = \frac{\text{cooc}(e, E)}{\sum_{e'} \text{cooc}(e', E)}.$$

We consider four settings of $\text{cooc}(e, E)$: (i) as maximum likelihood estimate, (ii) χ^2 hypothesis test, (iii) pointwise mutual information and (iv) log likelihood ratio; we briefly recall their details [21].

(i) Maximum likelihood estimate (MLE) uses the relative frequency of co-occurrences between e and E to determine the strength of their association:

$$\text{cooc}_{MLE}(e, E) = c(e, E)/c(E),$$

where $c(e, E)$ is the number of documents in which e and E co-occur and $c(E)$ is the number of documents in which E occurs.

(ii) The χ^2 hypothesis test determines if the co-occurrence of two entities is more likely than just by chance. A χ^2 test is given by:

$$\text{cooc}_{\chi^2}(e, E) = \frac{N \cdot (\, c(e, E) \cdot c(\overline{e}, \overline{E}) - c(e, \overline{E}) \cdot c(\overline{e}, E)\,)^2}{c(e) \cdot c(E) \cdot (\, N - c(e)\,) \cdot (\, N - c(E)\,)},$$

where N is the total number of documents, and $\overline{e}$, $\overline{E}$ indicate that e, E do not occur, respectively (i.e., $c(\overline{e}, \overline{E})$ is the number of documents in which neither e or E occurs).

(iii) Pointwise mutual information (PMI) determines the amount of information we gain if we observe e and E together. It is useful

[1]Evaluation script, qrels and additional resources are made publicly available at `http://ilps.science.uva.nl/resources/cikm2010-entity`.

Co-occ.	R-prec	R@100	R@2000	R@All
MLE	.0399	.2957	.7501	.9311
χ^2	**.1099**	**.3268**	**.8273**	.9311
PMI	.0244	.0981	.4888	.9311
LLR	.0399	.2957	.7184	.9311

Table 2: Results of the pure co-occurrence models.

to determine independence between entities, but of less value to determine how dependent two entities are. PMI is given by:

$$\text{cooc}_{PMI}(e, E) = \log c(e, E)/(c(e) \cdot c(E)).$$

(iv) Log likelihood ratio is another measure that determines dependence and is more reliable than PMI [10]. It is defined as:

$$\begin{aligned}\text{cooc}_{LLR}(e, E) \quad = \quad & 2(L(p_1, k_1, n_1) + L(p_2, k_2, n_2) \\ & -L(p, k_1, n_1) - L(p, k_2, n_2)),\end{aligned}$$

where $k_1 = c(e, E)$, $k_2 = c(e, \overline{E})$, $n_1 = c(E)$, $n_2 = N - c(E)$, $p_1 = k_1/n_1$, $p_2 = k_2/n_2$ and $p = (k_1 + k_2)/(n_1 + n_2)$, while $L(p, k, n) = k \log p + (n - k) \log(1 - p)$.

Results. Table 2 shows the results of the different estimation methods for the pure co-occurrence model. Out of the four methods, χ^2 is a clear winner while PMI performs worst on all metrics. MLE and LLR deliver very similar scores; their recall is comparable to that of χ^2, but they achieve much lower R-precision. All estimators return entities that co-occur at least once with the source entity, hence R@All is the same for all, just over 93%.

Analysis. The numbers presented in Table 2 demonstrate that simple co-occurrence statistics can achieve reasonable recall and can be used to obtain a candidate set of entities (e.g., top 2000) that can then be further examined by subsequent components in the pipeline. Fig. 3 (Top) shows the R@2000 scores of the methods per topic. For most topics at least one of the methods achieves high recall, with the exception of topic #4.

Unlike recall, R-precision scores are very low, suggesting that pure co-occurrence is not enough to solve the REF task. Fig. 3 (Bottom) shows that all methods score zero on R-precision on all but 4 topics. To identify the types of errors made, we take topic #17 (cf. Table 1) as an example and list the top 10 entities produced by our co-occurrence methods in Table 3.[2] Clearly, χ^2 finds relevant entities (bold) mixed with non-relevant entities that are not of the target type T (normal font). The other methods suffer more heavily from this type of error and fail to return any relevant entities in the top 10. We also see another type of error: entities, that are of the right type, but do not satisfy the target relation with the source entity. Note that one of the entities (indicated by †) is relevant, but not identified as such, as its Wikipedia page does not occur in the qrels.

Different co-occurrence methods display distinct characteristics in what they consider as strongly associated. MLE and LLR focus on popular entities; the top ranking entity, "Charitable Organization", occurs 5,271,075 times. The other extreme is demonstrated by PMI, which favors rare entities: the top ranking entity occurs 2 times and exclusively with the source entity. Finally, χ^2 performs well when entities co-occur frequently with the source entity and less with others; the top ranked entity occurs in 327 documents, in 187 cases together with the source entity, for the second best entity these numbers are 148 and 106, respectively.

As all methods and topics suffer from entities of the wrong type polluting the rankings, we address this next.

[2]We use topic #17 as a running example throughout the paper, to illustrate the impact of additional ranking and filtering components.

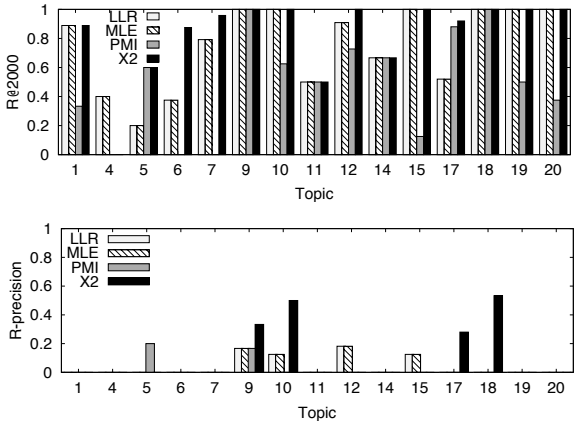


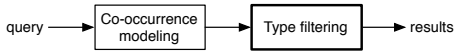
Figure 3: Per topic R@2000 (Top) and R-precision (Bottom) scores for the pure co-occurrence estimation methods.

	PMI	MLE/LLR	χ^2
1	Y'all (magazine)	Charitable organization	Iron Chef America
2	*Wayne Harley Brachman*	United States	**Paula Deen**
3	Veniero's	New York City	**Bobby Flay**
4	The Hungry Detective	2007	*Alton Brown*†
5	The FN Dish	2006	Fine Living
6	Super Suppers	NBC	**Rachael Ray**
7	Sugar Sugar, Inc	2008	Emeril Live
8	Stonewall Kitchen	Website	Unwrapped
9	*Raul Musibay*	Internet Movie Database	**Giada De Laurentiis**
10	Party Line with The Hearty Boys	American Broadcasting Company	Real Age

Table 3: Top 10 entities for topic #17. Relevant entities in bold, entities of the wrong type in roman, and entities of the right type but in the wrong relation in italics. MLE and LLR have the same top 10 ranking and are not displayed separately.

6. TYPE FILTERING

To combat the problem that results produced by the pure co-occurrence model are polluted by entities of the wrong type, we add a type filtering component on top of the pure co-occurrence model; this is indicated by the thick box in the figure below.



The challenge will be to maintain the high recall levels attained by the pure co-occurrence model while improving precision. Recall from (4), that type filtering is formalized as $P(T|e)$. The entity type filtering component $P(T|e)$ expresses the probability that an entity e is of the target type. Combined with the pure co-occurrence model, it yields the following model for ranking entities (we omit details of the derivation for brevity; it goes analogously to §3):

$$P(e|E, T) \propto P(e|E) \cdot P(T|e). \tag{5}$$

Estimation. In order to perform type filtering we exploit category information available in Wikipedia. We map each of the (input) entity types ($T \in \{PER, ORG, PROD\}$) to a set of Wikipedia categories ($\text{cat}(T)$) and we create a similar mapping from entities to categories ($\text{cat}(e)$). The former is created manually, while the latter is granted to us in the form of page-category assignments in Wikipedia (recall that Wikipedia pages correspond to entities). With these two mappings we estimate $P(T|e)$ as follows:

$$P(T|e) = \begin{cases} 1 & \text{if } \text{cat}(e) \cap \text{cat}^{L_n}(T) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Since the Wikipedia category structure is not a strict hierarchy and the category assignments are imperfect [9], we (optionally) expand

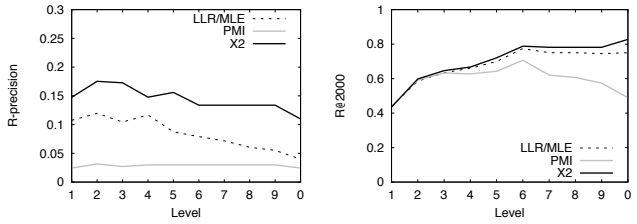


Figure 4: R-precision and R@2000 at increasing levels of category expansion.

the set of categories assigned to each target entity type T, hence write $\text{cat}^{L_n}(T)$, where L_n is the chosen level of expansion.

For the initial mapping of types to categories, $\text{cat}^{L_1}(T)$, we manually assign a number of categories to each type as in [19]. To the person type we map categories that end with "birth," "death," start with "People" and the category "Living People." To the organization type we map categories that start with "Organizations" or "Companies" and to the product type we map categories starting with "Products" or ending with "introductions." Next, we use the Wikipedia category hierarchy to expand this set by adding all direct child categories of the categories in L_1, to obtain our first expansion set L_2. We continue expanding the categories this way, one level at a time until no new categories are added.

While this particular form of type filtering is specific and tailored to Wikipedia, it is reasonable to assume that a named entity recognizer would provide us with high-level type information; therefore, it is not a limitation of the generalizability of our framework.

Results. By varying the expansion levels, we can optimize type filtering in two ways: for (R-)precision and for recall (R@2000). We first investigate the optimal levels of expansion for R-precision. Fig. 4 (Left) shows that R-precision increases when moving from level 0 (no filtering, shown on the right end of the plot) to level 2 expansion, but drops as the level of category expansion is further increased. This is in line with our expectation that an increasing number of categories allow more entities of the wrong type; because of the imperfection of the Wikipedia category structure, expansion results in the addition of many irrelevant categories.

As to recall, Fig. 4 (Right) shows R@2000 vs. level of expansion. R@2000 first increases and then decreases (PMI) or remains the same (MLE, LLR, and χ^2) as categories are expanded. At level 6 or beyond, the number of non-relevant entities allowed into the ranking is large enough to push relevant entities out of the top 2000. Uniformly applying category expansion down to the same level for all types is not necessarily optimal; some relevant entities of type organization are removed at expansion levels smaller than 6, while those of type person are only filtered out at level 1.

Table 4 shows the results of applying type filtering to the pure co-occurrence model optimized for precision (top rows) and for recall (bottom rows); relative changes are given w.r.t. the results in Table 2. We see an increase in R-precision for all methods. The best results when optimized for R-precision are achieved with χ^2, but we see large relative improvements for all methods in R-precision. Type filtering causes recall to drop sharply at low ranks; achieving max 60% R@2000 as opposed to 83% without filtering (cf. Table 2). The best R-precision scores averaged over all topics are achieved with type filtering at level 2 (L_2); this is the setting we will use when reporting scores optimized for R-precision.

As to the results optimized for recall, we find, again, that all methods improve both R-precision and R@100. The R@100 and R@2000 results suggest that χ^2 ranks relevant entities closer to the top 100 than the other methods. We find 79% of the relevant entities in the top 2000 and in total only 7% of the relevant entities

Co-occ.	R-Prec	R@100	R@2000	R@All
Optimized for Precision				
MLE	.1196 (+200%)	.3827 (+29%)	.5924 (-27%)	.5977 (-56%)
χ^2	**.1753** (+60%)	**.3976** (+22%)	**.5977** (-38%)	**.5977** (-56%)
PMI	.0316 (+30%)	.1920 (+96%)	.5910 (+21%)	.5977 (-56%)
LLR	.1196 (+200%)	.3827 (+29%)	.5857 (-23%)	.5977 (-56%)
Optimized for Recall				
MLE	.0791 (+98%)	.3915 (+32%)	.7740 (+3%)	.8667 (-7%)
χ^2	**.1338** (+22%)	**.5012** (+53%)	**.7881** (-5%)	.8667 (-7%)
PMI	.0298 (+22%)	.1344 (+37%)	.7065 (+45%)	.8667 (-7%)
LLR	.0791 (+98%)	.3915 (+32%)	.7740 (+8%)	.8667 (-7%)

Table 4: Results of type filtering with optimal level of filtering.

PMI	MLE/LLR	χ^2
1 Wayne Harley Brachman	Alton Brown[†]	**Paula Deen**
2 Kerry Vincent	**Rachael Ray**	**Bobby Flay**
3 Jacqui Malouf	**Bobby Flay**	**Alton Brown**
4 Glenn Lindgren	Chef	**Rachael Ray**
5 Geof Manthorne	**Paula Deen**	**Giada De Laurentiis**
6 Anna Pump	**Mario Batali**	**Mario Batali**
7 **Alexandra Guarnaschelli**	Oprah Winfrey	**Guy Fieri**
8 Kenji Fukui	George W. Bush	**Michael Symon**
9 Warren Brown	**Giada De Laurentiis**	**Cat Cora**
10 Tatsuo Itoh	**Emeril Lagasse**	Charles Scripps

Table 5: Top 10 entities for topic #17 with type filtering (L_2).

are lost by type filtering. We achieve the best recall scores with type filtering at level 6 (L_6); this is the value used for recall-optimized settings reported in the remainder of the paper.

By varying the level of expansion we can effectively aim either for R-precision or for R@2000, without hurting the other. This decision is likely to be made depending on whether this is the last component of the pipeline or results will be passed along for downstream processing. Optimizing category expansion levels for precision and recall carry the risk of overfitting, especially on a small topic set. Our aim with this tuning, however, is not to squeeze out the last bit of performance, but to demonstrate that type filtering can effectively be used to balance precision and recall. Two reasons reduce the risk of overfitting: (i) the target types are of a high level causing the granularity of category expansion to be of a coarse nature and (ii) the level of expansion is the same for all types.

Analysis. Table 5 shows the top 10 results for topic #17 after type filtering. We see that type filtering effectively removes entities of the wrong type from the ranking: all remaining entities are of type PER and no relevant entities were removed. Another type of error—entities of the right type but not engaged in the required relation R to the source entity E ("Chefs with a show on the Food Network")—, is now more prominent (see, e.g., *Oprah Winfrey* and *George W. Bush*). In §7 we address this type of error by adding context to the co-occurrence model and only admitting co-occurrences in contexts that display evidence of the required relation.

7. ADDING CONTEXT

To suppress entities that are of the right type T but that do not engage in the required relation R, we add an additional component: modeling contextual information (the thick box below):

Recall from (4) that the context of a co-occurrence model is captured as $P(R|E,e)$. Putting things, this is how we rank (§3):

$$P(e|E,T,R) \propto P(R|E,e) \cdot P(e|E) \cdot P(T|e). \quad (6)$$

Co-occ.	R-Prec	R@100	R@2000	R@All
Optimized for Precision				
MLE	**.2099** (+76%)	.4929 (+29%)	.5950 (0%)	.5977 (0%)
χ^2	.2094 (+19%)	.4631 (+16%)	**.5977** (0%)	.5977 (0%)
PMI	.0678 (+115%)	.2715 (+41%)	.5889 (-1%)	.5977 (0%)
LLR	.2032 (+70%)	**.4955** (+29%)	.5950 (+2%)	.5977 (0%)
Optimized for Recall				
MLE	**.1905** (+140%)	**.6221** (+60%)	.8344 (+8%)	.8667 (0%)
χ^2	.1798 (+34%)	.5708 (+14%)	**.8459** (+7%)	.8667 (0%)
PMI	.0678 (+127%)	.3313 (+147%)	.8315 (+18%)	.8667 (0%)
LLR	.1705 (+115%)	.5997 (+53%)	.8316 (+7%)	.8667 (0%)

Table 6: Results of the context dependent model.

Estimation. The $P(R|E, e)$ component is the probability that a relation is generated from ("observable in") the context of a source and candidate entity pair. We represent the relation between a pair of entities by a co-occurrence language model (θ_{Ee}), a distribution over terms taken from documents in which the source and candidate entities co-occur. By assuming independence between the terms in the relation R we arrive at the following estimation:

$$P(R|E, e) = P(R|\theta_{Ee}) = \prod_{t \in R} P(t|\theta_{Ee})^{n(t,R)}, \quad (7)$$

where $n(t, R)$ is the number of times t occurs in R. To estimate the co-occurrence language model θ_{Ee} we aggregate term probabilities from documents in which the two entities co-occur:

$$P(t|\theta_{Ee}) = \frac{1}{|D_{Ee}|} \sum_{d \in D_{Ee}} P(t|\theta_d), \quad (8)$$

where D_{Ee} denotes the set of documents in which E and e co-occur and $|D_{Ee}|$ is the number of these documents. $P(t|\theta_d)$ is the probability of term t within the language model of document d:

$$P(t|\theta_d) = \frac{n(t, d) + \mu \cdot P(t)}{\sum_{t'} n(t', d) + \mu}, \quad (9)$$

where $n(t, d)$ is the number of times t appears in document d, $P(t)$ is the collection language model, and μ is the Dirichlet smoothing parameter, set to the average document length in the collection [20].

Results. Table 6 shows the results of the context dependent model (including type filtering), optimized for precision (Top) and recall (Bottom); relative changes are w.r.t. the corresponding cells in Table 4. In both cases, R-precision and R@100 are substantially improved, while R@2000 and R@All remain the same or slightly improve. The best performing method across the board is MLE, but there is only a slight difference with the LLR and χ^2 scores. PMI achieved the largest relative improvements, but it still lags behind the other three methods for both R-precision and R@100.

Analysis. Looking at Table 7 we see that several entities have been replaced with others, "fresh" ones. Some that were in the "wrong" relation (i.e., *Oprah* and *Bush*, cf. Table 5) have been removed. For both MLE and LLR *Chef* and *Celebrity* are now returned at the top ranks; these entities are observed very frequently together with relation terms (and type filtering erroneously recognizes them as people). Some entities occur only in a handful of documents (<10), as a consequence of which very little evidence of the relation R can be found in their contexts (examples from the qrels include *Alexandra Guarnaschelli*, *Aida Mollenkamp*, *Daisy Martinez*). We observe a larger performance gain for the MLE and LLR based models than for χ^2. By introducing context, the result lists—consisting of frequent entities, favored by these models—are supplemented with entities that occur in suitable contexts. The entities found by the χ^2 model show a large overlap with those identified on the basis of context, hence limiting the performance gain.

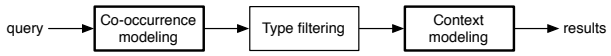
R	PMI	MLE	LLR	χ^2
1	Gennaro Contaldo	Chef	Chef	**Bobby Flay**
2	Asako Kishi	Celebrity	Celebrity	**Anne Burrell**
3	Yutaka Ishinabe	B. Smith	B. Smith	**Robert Irvine**
4	Karine Bakhoum	**Bobby Flay**	**Bobby Flay**	**Tyler Florence**
5	Masahiko Kobe	**Mario Batali**	**Mario Batali**	**Aaron McCargo, Jr.**
6	Tamio Kageyama	**Tyler Florence**	Bravo	**Mario Batali**
7	Toshiro Kandagawa	Bravo	**Rachael Ray**	Sunny Anderson†
8	Alpana Singh	**Rachael Ray**	**Tyler Florence**	**Guy Fieri**
9	Katie Lee Joel	**Robert Irvine**	**Paula Deen**	**Giada De Laurentiis**
10	Kazuko Hosoki	**Anne Burrell**	Alton Brown†	Kevin Brauch

Table 7: Top 10 entities for topic #17 after adding context.

These observations point to two issues with using Wikipedia as a corpus: (1) estimates for the pure co-occurrence models are unreliable and (2) the corpus is too small for constructing accurate context models, i.e., there is simply not enough textual material for certain entities. In §8 we address these problems by considering a larger corpus to improve our estimations of the pure co-occurrence model and to gather contexts for more robust context models.

8. IMPROVED ESTIMATIONS

We investigate how using a large corpus (CW-B, §4) for estimating our models can overcome the issue that for some entities their co-occurrences are limited to a small set of pages and that for some there is not enough context to be able to derive a robust language model. These changes affect two components of our pipeline:



Estimation. Using a large corpus for REF presents two challenges: NER on the entire corpus is time consuming and the shear number of entities becomes prohibitively large for any but the simplest of methods. To deal with these issues, we limit ourselves to a "working entity set" consisting of the top 2000 entities produced by the context dependent co-occurrence model (estimated on Wikipedia). We chose the entities returned for PMI without filtering as this produced the highest R@2000 (i.e., 87%). For our pure co-occurrence model we need, for each source-candidate entity pair, the number of documents in which they occur separately and the number of documents in which they co-occur (§5). We estimate these numbers by submitting the top 2000 entities as queries to an indexed version of CW-B, which returns the document IDs. We do the same for the source entities and then compare the document ID lists to find documents with co-occurrences. In order to estimate the context dependent model we consider only documents containing the source entity. We then create the co-occurrence model for a source-candidate entity pair by using the candidate as a query, effectively collecting all documents in which they co-occur.

Results. Table 8 shows the results for the co-occurrence models with estimates obtained from CW-B; relative changes in columns 2 and 3 are w.r.t. Table 4; those in columns 4 and 5 are w.r.t. Table 6. In the top left quadrant R-precision and R@100 of the pure co-occurrence model (optimized for precision) both improve over the same model using Wikipedia-based estimates for all methods: adding data solves the issue of sparse co-occurrences.

In the top right quadrant we see that the addition of context, using CW-B documents, further improves the χ^2 results, similar to what we saw when adding context in §7. In this case however, R-precision is worse than that achieved by the Wikipedia-based model for MLE, PMI, and LLR. In contrast, χ^2 shows a 25% improvement when adding CW-B documents. The models optimized for recall demonstrate a similar behavior; the pure co-occurrence model (bottom left) improves over the Wikipedia-based model, while the context dependent one does not, except for χ^2. For the χ^2 method, we

Co-occ.	Pure Co-Occurrence		Context Dependent	
	R-Prec	R@100	R-Prec	R@100
Optimized for Precision				
MLE	.1512 (+26%)	**.5423** (+42%)	.1898 (-11%)	**.5423** (+10%)
χ^2	**.2382** (+36%)	.4891 (+23%)	**.2623** (+25%)	.4747 (+3%)
PMI	.1363 (+331%)	.3545 (+85%)	.0649 (-8%)	.3137 (+16%)
LLR	.1540 (+29%)	.4947 (+29%)	.1767 (-15%)	.4873 (-2%)
Optimized for Recall				
MLE	.0799 (+1%)	**.5821** (+49%)	.0966 (-97%)	**.6982** (+12%)
χ^2	**.2281** (+70%)	.5474 (+9%)	**.2399** (+33%)	.5418 (-5%)
PMI	.0966 (+224%)	.3748 (+179%)	.0577 (-18%)	.3308 (0%)
LLR	.0793 (0%)	.5655 (+44%)	.0988 (-73%)	.6469 (+8%)

Table 8: Results for the context dependent model with filtering and estimations using the CW-B corpus.

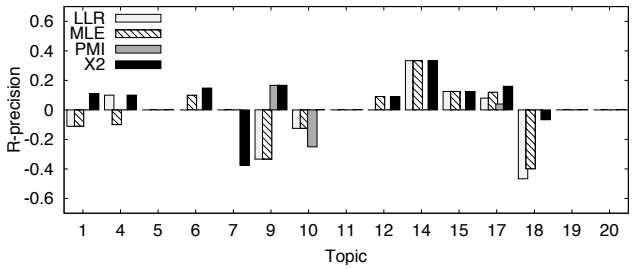


Figure 5: Differences in R-precision per topic; context dependent model using CW-B vs. Wikipedia. A negative score indicates greater precision for the Wikipedia-based model.

seem to have reached a good balance between precision and recall, continuing to improve R-precision with improvements or little effect on R@100. For the other methods, the picture is more diverse, especially for recall-optimized type filtering.

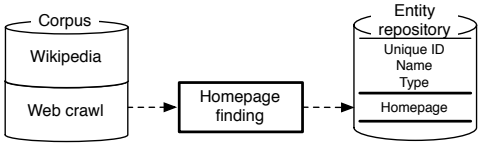
Analysis. Fig. 5 shows the difference per topic in R-precision of the context dependent model using either CW-B or Wikipedia; a negative score indicates higher R-precision for the Wikipedia-based model. Using Wikipedia documents greatly improves precision scores for three of the topics for MLE and LLR. As we look into these topics we see that the Wikipedia page of each source entity contains a full list of all the relevant entities (e.g., "members of the Beaux Arts Trio" and "members of Jefferson Airplane"), making them relatively easy, with external evidence likely to generate noise. However, the CW-B based model improves R-precision scores on a number of topics, which suggests that we can effectively use a larger corpus to handle a more diverse set of topics. In our running example (cf. Table 9) we now achieve a near perfect ranking for χ^2, MLE and LLR; PMI still finds only rare entities.

R	PMI	MLE	LLR	χ^2
1	Tamio Kageyama	Alton Brown†	Alton Brown†	**Bobby Flay**
2	Kazuko Hosoki	**Rachael Ray**	**Rachael Ray**	**Paula Deen**
3	Toshiro Kandagawa	**Bobby Flay**	**Bobby Flay**	Alton Brown†
4	Ron Siegel	**Mario Batali**	**Paula Deen**	**Michael Symon**
5	Mayuko Takata	**Paula Deen**	**Mario Batali**	**Giada De Laure.**
6	Asako Kishi	Chef	Chef	**Rachael Ray**
7	David Evangelista	**Cat Cora**	**Cat Cora**	**Mario Batali**
8	Dave Spector	**Emeril Lagasse**	**Emeril Lagasse**	**Cat Cora**
9	Kazushige Nagash.	**Michael Symon**	**Giada De Laure.**	**Guy Fieri**
10	Chua Lam	**Giada De Laure.**	**Michael Symon**	Kenji Fuku

Table 9: Top 10 entities with improved estimations for topic #17; some names truncated for layout reasons.

9. HOMEPAGE FINDING

Up to this point in the retrieval pipeline entities have been identified by their Wikipedia page. However, according the TREC Entity track, an entity is uniquely identified by its homepage, therefore we now focus on the homepage finding component in our architecture:



Approach. The 2009 Entity track allows up to three homepages and a Wikipedia page to be returned for each entity and judges pages as either primary[3], relevant or non-relevant. In this paper, we define homepage finding as the task of returning the primary homepage for an entity. Our approach combines language modeling based homepage finding and link-based approaches (see below), as a linear mixture with equal weights on the components.

Ranking homepages. We address homepage finding as a document retrieval problem, and employ a standard language modeling approach with uniform priors [31]; it ranks homepages according to the query likelihood: here, we use the name of the entity e_n as a query, $P(q = e_n|d)$. Successful approaches to named page and homepage finding use a combination of multiple document fields to represent documents [6, 23]. Following [23], we estimate $P(e_n|d)$ as a linear mixture of four components, constructed from the body, title, header and inlink fields. The parameters of the model are estimated empirically, see below.

Ranking links. Since our REF system identifies entities by their Wikipedia pages, it is natural to use the information on those pages for homepage finding; external links often contain a link to the entity's homepage [19, 30]. We, again, view this as a ranking problem and estimate the probability that document d is the homepage of entity e given a link e_{wl} on the entity's Wikipedia page: $P(d|e_{wl})$. We set this probability proportional to the position of the link among all external links on the Wikipedia page (pos(e_{wl})). Since we have to return "valid" homepages (i.e., that are present in CW-B), we perform an additional filtering step, and exlude URLs from our ranking which do not exist in CW-B.

We also employ a method based on DBpedia, which provides a list of entities with the URL of their homepage.[4] While these homepages may be more reliable than those found through the earlier external links strategy, the coverage of this method is limited. We set the probability of a homepage given a DBpedia URL, $P(d|e_{dl})$, to 1 if the URL exists in CW-B, and to 0 otherwise. To take advantage of the high quality, but sparse, data in DBpedia, while maintaining high coverage through external links in Wikipedia, we combine the external link and DBpedia strategies using a mixture model; for the sake of simplicity, we set equal weights to both components.

Evaluation. Before incorporating the homepage finding component into the end-to-end retrieval process, we evaluate its performance on its own. For this purpose we created a test set of homepage finding topics from TREC 2009 Entity qrels; we consider each entity with a primary homepage as a topic, and take the homepage as relevant document; topics and qrels are made available, see Fn. 1.

Parameter estimation (for the weights of the document fields in the mixture model) is done in two ways. The first uses the TREC 2002 Web track data [6]; while performance on the Web track's topics (MRR 0.69) is comparable to the best approaches at TREC

[3]A primary homepage is the main page about, and in control of, the entity, whereas a relevant page merely mentions the entity .

[4]Available at `http://wiki.dbpedia.org/Downloads33`.

	TREC evaluation				WP evaluation	
Method	P@10	#pri	nDCG@R	#rel	R-prec	R@100
Optimized for Precision (χ^2)						
(p1) Baseline (§8)	.2100	121	.1198	54	.2623	.5423
(p2) Improved typefiltering	.2350	157	.1399	62	.2959	.6017
(p3) Anchor based co-oc.	**.3000**	**174**	**.1562**	**76**	**.3473**	**.6667**
(p4) Adjusted judgements	.3900	186	.1966	78	.4759	.6869
Optimized for Recall (MLE)						
(r1) Baseline (§8)	.0800	171	.0880	105	.0966	.6982
(r2) Improved typefiltering	.1000	177	.1012	102	.1408	.7422
(r3) Anchor based co-oc.	**.1950**	**187**	**.1444**	**143**	**.2730**	**.7496**
(r4) Adjusted judgements	.3450	214	.2207	156	.4134	.8057
Best runs from the TREC Entity track						
KMR1PU [12]	**.4450**	137	**.2210**	115	**.5494**	**.5755**
ICTZHRun1 [43]	.3100	124	.1525	69	.3182	.4638
NiCTm2 [40]	.3050	124	.1689	98	.2721	.3820
tudpwkntop [30]	.2600	**144**	.1506	128	.2705	.5721
uogTrEpr [22]	.2350	135	.1760	**311**	.2945	.4536

Table 10: Comparison of our best runs and TREC results. Wikipedia pages are counted as primary homepages.

2002, this setting does not perform very well on CW-B (MRR 0.35). Our second parameter estimation method utilizes Wikipedia, using the page title as a name for the entity and considering external links with "official website" in their anchor text as homepages of that entity; this leads to a more acceptable performance of the mixture model on CW-B (MRR 0.47).

Turning to an evaluation of the homepage finding method, using the dedicated topics and judgments derived from TREC 2009 Entity qrels (Fn. 1), we find that, by itself, the DBpedia-based method results in very low scores (MRR 0.08), due its low coverage. Using external links from Wikipedia pages of entities we achieve a more acceptable score (MRR 0.44). Combining links from DBPedia and Wikipedia results only in minor improvements (MRR 0.45); this is not surprising given that DBPedia is extracted from Wikipedia. The best performance overall is achieved when the language modeling-based approach is combined with the two link-based approaches (MRR 0.62); this is the method that we use in the rest of the paper.

10. DISCUSSION

We now perform an end-to-end evaluation on the task specified at the TREC Entity track. According to the track's definition, up to 3 homepages and a Wikipedia page may be returned for each entity; each is judged on a 3-point scale (non-relevant, relevant or primary). We combine the pipeline developed in §5–§8 with the homepage finding component developed in §9. Table 10 presents the results. The *Baseline* row corresponds to our best performing run on CW-B (cf. §8). Note that we still only consider the 15 topics described in §4. Observe that our recall-oriented model (r1) outperforms other Entity track approaches in terms of the total number of primary pages found (#pri), while the precision-oriented model (p1) is in the top 6 in terms of precision ($P10$).

Next we take a look at how competitive our results are when we apply heuristic methods that were popular at the Entity track to our model. We experiment with two additional techniques.

Improved type filtering. Serdyukov and de Vries [30] use the high quality type definitions provided by the DBpedia ontology[5] to perform type filtering. We follow this approach and map the ontology categories "Person" and "Organization" to their respective topic target types (PER and ORG). We associate the class "Resource" with the product target type ($PROD$), as there is no specific product category in the ontology. In case an entity does not occur in the ontology, we fall back to our Wikipedia-based filtering (either precision- or recall-oriented), as described in §6. We incorporate this in the type filtering component of our model as follows:

$$P(T|e) = \begin{cases} 1 & \text{if ont}(e) \neq \emptyset \wedge T \cap \text{ont}(e) \neq \emptyset \\ 1 & \text{if ont}(e) = \emptyset \wedge \text{cat}(e) \cap \text{cat}^{L_n}(T) \neq \emptyset \\ 0 & \text{otherwise,} \end{cases}$$

where $T \in \{PER, ORG, PROD\}$ and $\text{ont}(e)$ returns the set of types for an entity in the DBpedia ontology.

The combination of our category based filtering approach with DBpedia based filtering has a positive effect on both precision and recall, Table 10 (p2 and r2). The two approaches complement each other as the category based filtering covers all entities, but is imprecise, while filtering based on the DBpedia ontology is precise, but only covers some of the entities in Wikipedia.

Anchor-based co-occurrence. Another approach employed at the Entity track [30, 40] is to only consider entities that link to, or are linked from, the Wikipedia page of the source entity. We view this as a special case of co-occurrence; its strength is proportional to the number of times source and target entities cross-link to each other on their corresponding Wikipedia pages. We estimate this anchor based co-occurrence as follows:

$$P_{anc}(e|E) = \lambda_a \cdot \frac{c(e, E_a)}{\sum_{e'} c(e', E_a)} + (1 - \lambda_a) \cdot \frac{c(E, e_a)}{\sum_{E'} c(E', e_a)},$$

where $c(e, E_a)$ is the number of times the candidate entity e occurs in the anchor text on the Wikipedia page of the input entity E, and $c(E, e_a)$ is the other way around. We incorporate this into the pure co-occurrence component (§5) as a sum with equal weights.

With the addition of the anchor-based co-occurrence we further improve our precision and recall scores; see Table 10 (p3 and r3). Anchor-based co-occurrence works well in this setting as for most topics the relevant entities occur as anchor texts on the page of the source entity and vice versa (e.g., topics #9 and #20).

Wikipedia-based evaluation. Another way to compare our model and those of other TREC participants is to use the Wikipedia based evaluation employed throughout the paper. From each participant's run we extract the Wikipedia fields and evaluate the number of primary Wikipedia pages for each topic in terms of R-precision and Recall@100. We observe that we outperform all but one of the other approaches in terms of R-precision (p3) and all approaches in terms of Recall@100 (r1, r2 and r3). The high precision achieved by the best performing team is due to their extensive use of heuristics, e.g., using a web search engine to collect relevant pages, crafting extraction patterns and exploiting lists and tables [12].

Adjusted judgements. Finally, the runs produced by our models are not official TREC runs and as such were not included in the assessment procedure; this might leave us with sparse judgments. Following standard TREC practice, non-judged documents are considered non-relevant—the resulting scores could therefore be an underestimation of our actual retrieval performance. To investigate how this affects our results we remove all entities for which there is no judgment available at all (neither primary, relevant or non-relevant, for neither the homepage or Wikipedia fields). We observe that only considering judged entities has a big affect on the precision and recall of our model (extended with anchor based co-occurrence and improved type filtering), see Table 10 (p4 and r4). In the precision oriented model 763 of the 6184 pages are judged (186 primary, 78 relevant). In the recall oriented model 1119 of the 6172 pages are judged (214 primary, 156 relevant). These num-

[5]Obtained from `http://wiki.dbpedia.org/Ontology`.

bers show that many of the returned entities have not been judged, impeding an assessment of the full potential of our models.

11. CONCLUSION

We examined an architecture for addressing the related entity finding (REF) task on a web corpus, where we focused on four core components: pure co-occurrence, type filtering, contextual information, and homepage finding. Initially we investigated the task on a smaller, less noisy corpus, using Wikipedia pages to uniquely identify entities. To identify a potential set of related entities we looked at four measures for computing co-occurrence and found that χ^2 performed best. An analysis showed that rankings of all methods were polluted by entities of the wrong type. We found that even a basic category based type filtering approach is very effective and that the level of category expansion can be tuned towards precision or recall. Furthermore, adding context improves both recall and precision by ensuring that source and target entities engage in the right relation. We then looked at the REF task in the setting of a web corpus and found that using a larger corpus improves the estimations of both co-occurrence and context models. To conform to the official REF task we used a homepage finding component to map the Wikipedia entity representation to a homepage and found that our framework achieves decent precision and very high recall scores compared to other approaches on the official task. Finally, we found that our model can effectively incorporate additional heuristics that lead to state-of-the-art performance.

Acknowledgements. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802, and partially by the Center for Creation, Content and Technology (CCCT).

12. REFERENCES

[1] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR '06*, pages 43–50, 2006.
[2] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Proc. and Man.*, 45(1):1–19, 2009.
[3] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *TREC '09*, 2009.
[4] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of TREC 2005. In *TREC 2005*, 2005.
[5] ClueWeb09. The ClueWeb09 dataset, 2009. URL: `http://boston.lti.cs.cmu.edu/Data/clueweb09/`.
[6] N. Craswell and D. Hawking. Overview of the TREC-2002 Web Track. In *TREC '02*, pages 86–95, 2002.
[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 Enterprise Track. In *TREC '05*, 2005.
[8] N. Craswell, G. Demartini, J. Gaugaz, and T. Iofciu. L3S at INEX2008. In Geva et al. [14], pages 253–263.
[9] A. de Vries et al. Overview of the INEX 2007 Entity Ranking Track. In Fuhr et al. [13], pages 245–251.
[10] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Comp. Ling.*, 19(1):61–74, 1993.
[11] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR 2007*, pages 418–430, 2007.
[12] Y. Fang et al. Entity retrieval with hierarchical relevance model. In *TREC '09*, 2009.
[13] N. Fuhr, J. Kamps, M. Lalmas, and A. Trotman, editors. *Focused access to XML documents: INEX 2007*. Springer, 2008.
[14] S. Geva, J. Kamps, and A. Trotman, editors. *Advances in Focused Retrieval: INEX 2008*. Springer, 2009.
[15] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *ACL '04*, 2004.
[16] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Comp. Ling. '92*, pages 539–545, 1992.
[17] J. Jiang et al. Adapting language modeling methods for expert search to rank Wikipedia entities. In Geva et al. [14], pages 264–272.
[18] Y. Jing and W. B. Croft. An association thesaurus for information retrieval. In *RIAO '94*, pages 146–160, 1994.
[19] R. Kaptein, M. Koolen, and J. Kamps. Result diversity and entity ranking experiments. In *TREC '09*, 2009.
[20] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*, 2001.
[21] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
[22] M. McCreadie, C. Macdonald, I. Ounis, J.Peng, and R. L. T. Santos. University of Glasgow at TREC 2009. In *TREC '09*, 2009.
[23] P. Ogilvie and J. P. Callan. Combining document representations for known-item search. In *SIGIR '03*, pages 143–150, 2003.
[24] J. Pamarthi, G. Zhou, and C. Bayrak. A journey in entity related retrieval for TREC 2009. In *TREC '09*, 2009.
[25] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *CIKM '07*, pages 731–740, 2007.
[26] H. Raghavan, J. Allan, and A. Mccallum. An exploration of entity models, collective classification and relation description. In *LinkKDD '04*, 2004.
[27] D. Ravichandran and E. H. Hovy. Learning surface text patterns for a question answering system. In *ACL '02*, pages 41–47, 2002.
[28] E. Riloff. Automatically generating extraction patterns from untagged text. In *AAAI, Vol. 2*, pages 1044–1049, 1996.
[29] S. Schlobach, M. Olsthoorn, and M. de Rijke. Type checking in open-domain question answering. In *ECAI '04*, 2004.
[30] P. Serdyukov and A. de Vries. Delft university at the TREC 2009 Entity Track: Ranking wikipedia entities. In *TREC '09*, 2009.
[31] F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99*, pages 316–321, 1999.
[32] T. Tsikrika et al. Structured document retrieval, multimedia retrieval, and entity ranking using PF/Tijah. In Fuhr et al. [13], pages 306–320.
[33] A.-M. Vercoustre, J. Pehcevski, and J. A. Thom. Using Wikipedia categories and links in entity ranking. In Fuhr et al. [13].
[34] E. M. Voorhees. Overview of the TREC 2002 Question Answering Track. In *TREC '02*, pages 115–123, 2002.
[35] E. M. Voorhees. The TREC-8 question answering track report. In *TREC '99*, pages 77–82, 1999.
[36] E. M. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In *SIGIR '02*, pages 316–323, 2002.
[37] V. G. V. Vydiswaran, K. Ganesan, Y. Lv, J. He, and C. Zhai. Finding related entities by retrieving relations. In *TREC '09*, 2009.
[38] Z. Wang, D. Liu, W. Xu, G. Chen, and J. Guo. BUPT at TREC 2009: Entity Track. In *TREC '09*, 2009.
[39] W. Weerkamp, J. He, K. Balog, and E. Meij. A generative language modeling approach for ranking entities. In Geva et al. [14].
[40] Y. Wu and H. Kashioka. NiCT at TREC 2009: Employing three models for Entity Ranking Track. In *TREC '09*, 2009.
[41] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM TOIS*, 18:79–112, 2000.
[42] Q. Yang, P. Jiang, C. Zhang, and Z. Niu. Experiments on related entity finding track at TREC 2009. In *TREC '09*, 2009.
[43] H. Zhai, X. Cheng, J. Guo, H. Xu, and Y. Liu. A novel framework for related entities finding: ICTNet at TREC 2009. In *TREC '09*, 2009.
[44] W. Zheng, S. Gottipati, J. Jiang, and H. Fang. UDEL/SMU at TREC 2009 Entity Track. In *TREC '09*, 2009.
[45] J. Zhu, D. Song, and S. Rüger. Integrating document features for entity ranking. In Fuhr et al. [13], pages 336–347.