Document Retrieval Using Entity-Based Language Models

Hadas Raviv
Technion, Israel
hadasrv@tx.technion.ac.il

Oren Kurland
Technion, Israel
kurland@ie.technion.ac.il

David Carmel
Yahoo Research, Israel
david.carmel@ymail.com

ABSTRACT

We address the ad hoc document retrieval task by devising novel types of entity-based language models. The models utilize information about single terms in the query and documents as well as term sequences marked as entities by some entity-linking tool. The key principle of the language models is accounting, simultaneously, for the uncertainty inherent in the entity-markup process and the balance between using entity-based and term-based information. Empirical evaluation demonstrates the merits of using the language models for retrieval. For example, the performance transcends that of a state-of-the-art term proximity method. We also show that the language models can be effectively used for cluster-based document retrieval and query expansion.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

Keywords: document retrieval; entity-based language models

1. INTRODUCTION

Most ad hoc document retrieval methods compare query and document representations. To address the potential vocabulary mismatch between a short query and documents relevant to the query, various semantic document-query similarity measures have been proposed [28].

Specifically, there is a growing body of work on retrieval methods that utilize information about *entities* in a repository (e.g., Wikipedia or Freebase) which appear in queries and documents (e.g., [46, 35, 39, 7, 13, 31, 45, 29, 33, 44]). Most of these methods expand the query with terms or entities related to those appearing (or marked) in it [46, 35, 39, 7, 13, 31, 45, 29]; other methods project queries and documents onto a latent or explicit entity space [14, 33, 44].

In this paper we take a step back, and address a more fundamental challenge regarding the use of entity-based information for document retrieval. We study whether using *surface level* entity-based query and document representations can help to improve retrieval effectiveness. By "surface level" we refer to representations based only on terms in the text and *markups* of entities in it, along with raw corpus-based occurrence statistics. This is in contrast to expansion-based and projection-based representations that utilize also terms and entities related to those (marked) in the text and which often use auxiliary information about entities from the entity repository; e.g., textual descriptions of entities, entities' categories and inter-entity relations [46, 35, 39, 7, 13, 31, 45, 29, 33, 44]. Put in simpler words, the question we address is *whether the markup of entities in a query and documents is, by itself, sufficient information for improving retrieval effectiveness.*

The reason for addressing the question just posed is two fold. First, it will shed light on the effectiveness of using entities in their most basic capacity; that is, special tokens marked in queries and documents. Indeed, findings in past work on ad hoc retrieval regarding the merits of using surface level entity-based representations are inconclusive [16, 42, 47, 3, 14]. Second, such representations can be naturally used in existing retrieval approaches and tasks to improve performance; e.g., query expansion and cluster-based document retrieval as we show in this paper.

There are various potential merits in using surface level entity-based representations. For example, these can help to cope with the vocabulary mismatch problem; e.g., the entity *United States of America* can have different expressions in the text, including, "U.S.", "USA", "United States" and more. Furthermore, expressions of entities in the text are variable-length n-grams that bear semantic meaning. Thus, entities can be used for effective modeling of term proximity information which goes beyond using fixed-length n-grams.

An important challenge in inducing entity-based representations is accounting for the uncertainty inherent in the entity-markup process (a.k.a. entity linking); that is, associating term sequences with entities in a repository. Specifically, a term sequence can potentially be associated with multiple entities; e.g., the term "Lincoln" can be associated with the U.S. president, the car, the 2012 movie, etc. The uncertainty in entity linking has significant impact on retrieval effectiveness as we show in this paper.

We present novel types of entity-based language models which consider *both* single terms in the text as well as term sequences marked as entities by an existing entity-linking tool. These language models are induced from the query and documents in the corpus and serve for retrieval in the language modeling framework. The main novelty of these language models is accounting, simultaneously, for (i) the uncertainty in entity linking — specifically, the confidence levels of entity markups; and, (ii) the balance between using

SIGIR '16, July 17-21, 2016, Pisa, Italy

 ISBN 978-1-4503-4069-4/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2911451.2911508

term-based and entity-based information. We demonstrate the importance of accounting for the mutual effects of these two aspects. For example, we show that using high recall entity markup, which is quite noisy, can help to significantly improve retrieval effectiveness if the noise is "balanced" by sufficient utilization of term-based information.

Empirical evaluation demonstrates the merits of using our entity-based language models for retrieval. The performance significantly transcends that of a state-of-the-art term proximity method: the sequential dependence model (SDM) [36, 19]. Integrating the language models with SDM yields further performance improvements. The language models are also effective for two additional retrieval paradigms: cluster-based document retrieval and query expansion.

2. RELATED WORK

The work most related to ours is that on devising surface level entity-based document and query representations for document retrieval [21, 16, 42, 47, 3, 41, 14]. The findings about the merits of these representations have been inconclusive. The few cases where the representations were shown to be somewhat effective for retrieval were when entity markups were devised in extreme care and were of very high quality [47, 3, 14]. In contrast to this past work that focused on vector space models, we demonstrate the clear merits of using our entity-based language models for retrieval. Also, in contrast to previously proposed representations [21, 16, 42, 47, 3, 41, 14], our language models account simultaneously for the uncertainty in the entity-markup process, and the balance between using term-based and entity-based information. Consequently, a highly important aspect that further differentiates our approach from past work is the effective utilization of high recall, noisy, entity markups.

There is work on query expansion using entity-based information [43, 34, 30, 40, 8, 10, 18, 46, 6, 20, 7, 35, 39, 13, 31, 29, 45] and on projecting queries and documents onto an entity space to compare them [14, 33, 44]. There are two fundamental differences between all this past work and ours which focuses on surface level entity-based query and document representations. First, in these past methods, queries and documents are represented by external terms and entities which they do not contain[1]. Our surface level representations do not utilize such expansions. Second, auxiliary information about entities from the entity repository (e.g., textual descriptions of entities and their interrelations) is utilized in this past work, but not in our representations[2].

We show that our entity-based language models can be used to create effective expanded query forms by "plugging" them into an *existing* query expansion method: the relevance model [26, 1]. The resultant approach, which *simultaneously* expands the query with both terms and entities, is conceptually reminiscent of some methods recently proposed by Dalton et al. [13]. In their work, queries are expanded, independently, using terms and entities. The retrieval scores attained by using multiple term-only and entity-only expanded query forms are fused using a learning-to-rank method [13]. We show that our language models can be used to further improve the effectiveness of such expansion-based approaches by improving the quality of the pseudo relevant document list used for query expansion.

We also demonstrate the merits of using our language models for cluster-based document retrieval. Using entity-based representations for this task is novel to this study.

In some studies, concepts (entities) in verbose queries were automatically weighted [2, 22, 4, 5]. In contrast to our approach, weights (confidence levels) of entities in documents were not accounted for. We demonstrate the importance of accounting for the confidence level of entity markups in both queries and documents. Further tuning of entities' weights in our proposed language models, using some of these approaches [2, 22, 4, 5], is interesting future work.

There are language models that integrate word phrases and named entities based on their association with predefined classes [27, 23]. In contrast to our language models, which are not based on such classes, these language models were not designed and used for document retrieval.

3. RETRIEVAL FRAMEWORK

In what follows we present ad hoc document retrieval methods that rank documents in a corpus D in response to query q. The methods utilize information about entities mentioned in the query and in documents.

To mark entities in texts, we use *some* entity-linking tool that utilizes a repository (e.g., Wikipedia or Freebase) where entities have unique IDs. The entity-linking tool takes as input a text, query or document in our case, and marks variable length sequences of terms as *potential* entities in the repository. The entity markup of a term sequence is composed of entity ID and a confidence level in $[0, 1]$. The confidence level reflects the likelihood that the term sequence corresponds to the entity. The confidence level relies on the term sequence and its context; e.g., its neighboring terms or other term sequences marked as entities [15, 38]. Using high confidence level results in high precision entity markup while low confidence level results in high recall.

We assume that each position in a given text can be part of at most a single term sequence that is marked as an entity; i.e., the entity markups do not overlap. A specific occurrence of a term sequence in a text cannot be marked with more than one entity. Yet, a term sequence can appear several times in a text with different entity markups as the markups depend on the context of the sequence. Details of the entity linking tools we use are provided in Section 4.1.

The retrieval methods we present in Section 3.2 use entity-based query and document language models. We now turn to define these language models.

3.1 Entity-based language models

We define unigram entity-based language models over a token space $\mathcal{T}$; i.e., tokens are generated by the language model independently of each other. The token space,

$$\mathcal{T} \stackrel{def}{=} \mathcal{V} \cup \mathcal{E} \quad (1)$$

is composed of the set $\mathcal{V}$ of all terms in the corpus D and the set $\mathcal{E}$ of entities in the entity repository which were marked at least once in a document in D with *any* confidence level.

[1]Xiong and Callan [44] found that representing queries using only entities marked in them is of merit for their learning-to-rank approach. However, features describing the query-entities relations rely on auxiliary information from the entity repository that is not used by our methods.

[2]The entity-linking process could use auxiliary information from the entity repository. However, our proposed representations utilize the entity markups simply as tokens with confidence levels, and do not use auxiliary information.

The language models we devise rely on a definition of *pseudo counts* for tokens. Two definitions of pseudo counts will be presented in Sections 3.1.1 and 3.1.2. Let $pc(t,x)$ be the pseudo count of token t ($\in \mathcal{T}$) in the text or text collection x. We define the *pseudo length* of x as:

$$pl(x) \stackrel{def}{=} \sum_{t \in \mathcal{T}: pc(t,x)>0} pc(t,x).$$

The maximum likelihood estimate (MLE) of token t ($\in \mathcal{T}$) with respect to x is:

$$\theta_x^{MLE}(t) \stackrel{def}{=} \frac{pc(t,x)}{pl(x)}. \quad (2)$$

The MLE can be smoothed using Dirichlet priors [49]:

$$\theta_x^{Dir}(t) \stackrel{def}{=} \frac{pc(t,x) + \mu \theta_D^{MLE}(t)}{pl(x) + \mu}; \quad (3)$$

μ is a smoothing parameter.

We next describe two types of language models defined over $\mathcal{T}$ and induced using Equations 2 and 3. The language models differ by the definition of pseudo counts for tokens.

3.1.1 Hard confidence-level thresholding

The hard confidence-level thresholding language model, **HTLM** in short, is based on *fixing* a threshold τ ($\in [0,1]$) for entity markups. Entity-based information is used only for entity markups with confidence level $\geq \tau$. In contrast, *every* term occurrence in a text, including those in entity markups with a confidence level $< \tau$, is accounted for.

To formally define a HTLM using Equations 2 and 3, we have to define pseudo counts for tokens from $\mathcal{T}$ in a text or text collection x. To that end, we lay down a few definitions. If t ($\in \mathcal{T}$) is a term, then $c_{term}(t,x)$ is the number of occurrences of t in x. Let $\mathcal{M}(x)$ denote the set of all entity markups in x; i.e., all occurrences of term sequences in x that were marked as entities with some confidence level. For a markup m ($\in \mathcal{M}(x)$), $E(m)$ is the entity and $\rho(m)$ is the confidence level. The equivalence relation $t_1 \equiv t_2$ holds iff the entity tokens t_1 and t_2 are identical (i.e., have the same ID). The pseudo count of t ($\in \mathcal{T}$) in x is based on (i) the raw count of t in x if t is a term; and, (ii) the number of entity markups of t in x with a confidence level $\geq \tau$ if t is an entity. Formally,

$$pc_{HTLM;\tau}(t,x) \stackrel{def}{=} \quad (4)$$
$$\begin{cases} \lambda c_{term}(t,x) & \text{if } t \in \mathcal{V}; \\ (1-\lambda) \sum_{m \in \mathcal{M}(x): E(m) \equiv t} \delta[\rho(m) \geq \tau] & \text{if } t \in \mathcal{E}; \end{cases}$$

λ ($\in [0,1]$) is a free parameter which controls the relative importance attributed to term and entity tokens; δ is Kronecker's delta function: for statement s, $\delta[s] = 1$ if s is true and $\delta[s] = 0$ otherwise.

We note that using a Dirichlet smoothed HTLM (i.e., using Equation 4 in Equation 3) can still result in assigning zero probability to some tokens in $\mathcal{T}$. These are entities with no corresponding markup of a term sequence in the corpus with confidence level $\geq \tau$. We re-visit this point below.

If we set $\lambda = 1$ in Equation 4, then the resultant HTLM reduces to a standard unigram term-based language model. Setting $\lambda = 0$ results in **HTEntLM** which is a unigram language model that assigns non-zero probability *only* to entities: if the MLE from Equation 2 is used, then these are the entities with at least one markup in x with a confidence level $\geq \tau$; if the Dirichlet smoothed language model is used (Equation 3), then these are the entities with at least one markup in the corpus with a confidence level $\geq \tau$.

3.1.2 Soft confidence-level thresholding

A potential drawback of HTLM is committing to a specific threshold τ for entity markups. That is, information about entity markups with confidence level lower than τ is ignored. Furthermore, all entity markups with confidence level $\geq \tau$ are counted equally as their confidence levels are ignored.

Thus, we now turn to present a soft confidence-level thresholding language model, **STLM**. STLM accounts for all markups of an entity and weighs them by the corresponding confidence levels. Specifically, the pseudo count of t ($\in \mathcal{T}$) in the text or text collection x is defined as:

$$pc_{STLM}(t,x) \stackrel{def}{=} \begin{cases} \lambda c_{term}(t,x) & \text{if } t \in \mathcal{V}; \\ (1-\lambda) \sum_{m \in \mathcal{M}(x): E(m) \equiv t} \rho(m) & \text{if } t \in \mathcal{E}; \end{cases} \quad (5)$$

λ ($\in [0,1]$) is a free parameter that, as in HTLM, controls the relative importance attributed to term and entity tokens. Thus, STLM addresses the uncertainty inherent in the entity linking process by using *expected* entity occurrence counts; the corresponding confidence levels serve for occurrence probabilities. These expected counts are then integrated with deterministic term counts.

If we set $\lambda = 1$ in Equation 5, then STLM reduces to a standard unigram term-based language model as was the case for HTLM. Setting $\lambda = 0$ results in **STEntLM**. This language model assigns a non-zero probability only to entities that have at least one markup (with any confidence level) in x when using the MLE (Equation 2) or in the corpus when using the Dirichlet smoothed language model (Equation 3). We note that in contrast to the case for HTLM, there is no token in $\mathcal{T}$ that is assigned a zero probability by a Dirichlet smoothed STLM.

3.2 Retrieval models

We rank document d by the cross entropy between the language models induced from the query (q) and d [25]:

$$CE(\theta_q \mid\mid \theta_d) = -\sum_{t \in \mathcal{T}} \theta_q(t) \log \theta_d(t); \quad (6)$$

higher values correspond to decreased similarity.

Equation 6 is instantiated using the entity-based language models described in Section 3.1. Following common practice [48], we use an unsmoothed maximum likelihood estimate for the query language model (Equation 2) and a Dirichlet smoothed document language model (Equation 3). We obtain four retrieval methods : **HT**[3], **HTOEnt**, **ST** and **STOEnt**[4], which utilize the HTLM, HTEntLM, STLM and

[3]In HT, the *same* confidence-level threshold, τ_d, is used for all documents; the query threshold, τ_q, can be different from τ_d. Hence, an entity token assigned a non-zero probability by θ_q could be assigned a zero probability by θ_d; e.g., an entity marked in q with a confidence level $\geq \tau_q$ but with no markup in the corpus with confidence level $\geq \tau_d$. In these cases, we zero the probability assigned to the entity token by θ_q to avoid a $\log 0$ in Equation 6. This is common practice in addressing term tokens that appear in a query but not in any document in the corpus.

[4]HTOEnt and STOEnt rely only on entity tokens. If all entities in $\mathcal{E}$ are assigned a zero probability by the unsmoothed

Table 1: TREC data used for experiments.

corpus	# of docs	data	queries
AP	242,918	Disks 1-3	51 – 150
ROBUST	528,155	Disks 4-5 (-CR)	301 – 450, 601 – 700
WT10G	1,692,096	WT10g	451 – 550
GOV2	25,205,179	GOV2	701 – 850
ClueB ClueBF	50,220,423	ClueWeb09 (Cat. B)	1 – 200

STEntLM language models, respectively. HT and ST utilize entity and term tokens, while HTOEnt and STOEnt utilize only entity tokens, hence the "O" in the methods names.

3.2.1 Score-based fusion

The HTLM and STLM language models integrate term-based and entity-based information at the *language model level*. Hence, the query-document comparison in Equation 6 simultaneously accounts for the appearance of the query terms and entities in a document.

An alternative approach is integrating term and entity information at the *retrieval score level*. Inspired by approaches in the vector-space model [42], and in work on using a latent entity space [33], we consider a method that fuses document retrieval scores produced by utilizing, *independently*, term-only (θ_x^{term}) and entity-only (θ_x^{ent}) language models induced from text x. Document d is scored by:

$$\lambda CE\left(\theta_q^{term} \mid\mid \theta_d^{term}\right) + (1-\lambda) CE\left(\theta_q^{ent} \mid\mid \theta_d^{ent}\right); \quad (7)$$

the λ parameter balances the score fusion[5]. The query language models are unsmoothed maximum likelihood estimates (Equation 2) and the document language models are Dirichlet smoothed (Equation 3).

Instantiating Equation 7 with an entity-only language model, HTEntLM or STEntLM, and with a standard unigram term-based language model yields the **F-HT** and **F-ST** methods, respectively. These are conceptually highly similar to the HT and ST methods which integrate term-based and entity-based information at the language-model level. However, HT and ST use a single smoothing parameter for both term and entity tokens (see Equation 3) while F-HT and F-ST can use a different smoothing parameter for each as they utilize separately term-only and entity-only language models. We could have used different smoothing parameters for entity and term tokens under the same language model, e.g., by applying term-specific smoothing [17], but we leave this exploration for future work.

4. EVALUATION

4.1 Experimental setup

Experiments were conducted using the TREC datasets specified in Table 1. AP and ROBUST are mostly composed of news articles. WT10G is a small, noisy, Web collection. GOV2 is a much larger Web collection composed of high quality pages crawled from the .gov domain. ClueB is the English part of the Category B of the ClueWeb 2009 Web collection. ClueBF was created from ClueB by filtering from rankings suspected spam documents: those assigned a score below 50 by Waterloo's spam classifier [11].

Data processing. Titles of TREC topics served for queries. Tokenization and Porter stemming were applied using the Lucene toolkit (lucene.apache.org) which was used for experiments. Stopwords on the INQUERY list were removed from queries but not from documents.

Unless otherwise specified, the **TagMe** entity-linking tool (tagme.di.unipi.it) is used to annotate queries and documents. TagMe uses Wikipedia (a July 2014 dump) as the entity repository, and was shown to be highly effective and efficient in comparison to other publicly available entity-linking systems [12]. In Section 4.2.1 we also show the effectiveness of our methods using the **Wikifier** entity-linking tool[6] [9, 12]. Wikifier was applied with an efficient configuration claimed to yield baseline entity linking effectiveness.

TagMe and Wikifier cannot process very long texts. Thus, we split documents into non-overlapping term-window passages. We terminate a passage at the first space that appears at least 500 characters after the beginning of the previous passage. We let the tools mark the passages independently. The tools are applied on the non-stemmed and non-stopped queries and documents. Entity markup of a term sequence includes an entity ID and a confidence level (in $[0, 1]$). We scan each text left to right and remove overlapping entity markups so that each position can be part of at most a single markup. If two markups overlap, we select the one with the higher confidence level. We break ties of confidence levels by selecting the markup which starts at the leftmost position.

Baselines. We use standard term-based unigram language model retrieval [25], denoted **TermsLM**, for reference. This is a special case of the HT, ST, F-HT and F-ST methods with $\lambda = 1$. Documents are ranked by the cross entropy between the unsmoothed (MLE) query language model and Dirichlet smoothed document language models.

The **HTCon** method is a special case of HT with $\lambda = 0.5$ and $\tau_q = \tau_d = 0$ (τ_q and τ_d are the query and document thresholds, respectively). HTCon accounts *uniformly* for all entity mentions, and attributes the same importance to term and entity tokens. HTCon is conceptually reminiscent of methods representing documents and queries using concepts (e.g., from Wordnet) by concatenating with equal weights term-based and concept-based vector-space representations [41, 16, 42]. Accordingly, we consider **F-HTCon**: a special case of F-HT with $\lambda = 0.5$ and $\tau_q = \tau_d = 0$.

Additional baseline is the state-of-the-art sequential dependence model, **SDM**, from the Markov Random Field framework which utilizes term proximities [36, 19]. The comparison with SDM, and its integration with our STLM is presented in Section 4.2.3.

Evaluation measures and free-parameters. Mean average precision at cutoff 1000 (MAP), precision of the top 10 documents (p@10) and NDCG@10 (NDCG) serve as evaluation measures. Statistically significant performance differences are determined using the two-tailed paired t-test with a 95% confidence level.

query language model, then no documents are retrieved. This can happen for example when inducing HTEntLM from the query with a high confidence-level threshold or inducing a STEntLM from a query which has no entity markups.

[5]The λ in the score-based fusion model has a conceptually similar role to that of λ in STLM and HTLM: balancing the use of term-based and entity-based information.

[6]cogcomp.cs.illinois.edu/page/demo_view/Wikifier

Table 2: Comparison of methods instantiated from Equation 6 using term-only (TermsLM) and entity-based language models. Bold: the best result in a row. 't', 'h', 'o', 'c' and 's' mark statistically significant differences with TermsLM, HT, HTOEnt, HTCon and ST, respectively.

		TermsLM	HT	HTOEnt	HTCon	ST	STOEnt
AP	MAP	20.9	23.1^{t}	$15.6^{t,h}$	22.5_{o}	$\mathbf{23.5^{t}_{o,c}}$	$17.5^{t,h}_{o,c,s}$
	p@10	39.1	$\mathbf{44.2^{t}}$	36.0^{h}	43.4^{t}_{o}	43.8^{t}_{o}	$38.3^{h}_{c,s}$
	NDCG	40.4	45.3^{t}	37.6^{h}	44.7^{t}_{o}	$\mathbf{45.5^{t}_{o}}$	$39.6^{h}_{c,s}$
ROBUST	MAP	25.0	$\mathbf{28.1^{t}}$	$19.1^{t,h}$	27.4^{t}_{o}	$\mathbf{28.1^{t}_{o,c}}$	$21.4^{t,h}_{o,c,s}$
	p@10	42.2	$\mathbf{45.5^{t}}$	$35.7^{t,h}$	45.0^{t}_{o}	45.3^{t}_{o}	$38.0^{t,h}_{o,c,s}$
	NDCG	43.5	$\mathbf{47.1^{t}}$	$36.9^{t,h}$	46.3^{t}_{o}	46.9^{t}_{o}	$39.2^{t,h}_{o,c,s}$
WT10G	MAP	19.1	21.9^{t}	$13.3^{t,h}$	21.4^{t}_{o}	$\mathbf{22.9^{t,h}_{o,c}}$	$16.7^{h}_{o,c,s}$
	p@10	27.3	30.4^{t}	$21.6^{t,h}$	30.5^{t}_{o}	$\mathbf{31.6^{t}_{o}}$	$25.3^{h}_{o,c,s}$
	NDCG	30.3	32.7	$21.2^{t,h}$	32.1_{o}	$\mathbf{34.3^{t}_{o,c}}$	$25.4^{h}_{o,c,s}$
GOV2	MAP	29.6	32.1^{t}	$18.0^{t,h}$	30.6^{h}_{o}	$\mathbf{32.2^{t}_{o,c}}$	$20.7^{t,h}_{o,c,s}$
	p@10	53.9	57.3^{t}	$39.4^{t,h}$	56.8_{o}	$\mathbf{57.7^{t}_{o}}$	$44.0^{t,h}_{o,c,s}$
	NDCG	44.8	47.4^{t}	$32.7^{t,h}$	46.9_{o}	$\mathbf{47.9^{t}_{o}}$	$35.7^{t,h}_{o,c,s}$
ClueB	MAP	17.1	18.7^{t}	$14.0^{t,h}$	18.5_{o}	$\mathbf{19.5^{t}_{o}}$	$14.0^{t,h}_{c,s}$
	p@10	22.7	25.9^{t}	23.9	26.7^{t}_{o}	$\mathbf{27.4^{t}}$	24.1
	NDCG	16.5	18.7^{t}	18.3	19.2^{t}	$\mathbf{19.3^{t}}$	17.5
ClueBF	MAP	18.8	$\mathbf{20.5^{t}}$	$14.4^{t,h}$	19.9_{o}	20.3^{t}_{o}	$14.4^{t,h}_{c,s}$
	p@10	33.6	37.9^{t}	29.2^{h}	$\mathbf{38.2^{t}_{o}}$	37.9^{t}_{o}	$30.6^{h}_{c,s}$
	NDCG	24.3	$\mathbf{28.4^{t}}$	22.2^{h}	$\mathbf{28.4^{t}_{o}}$	27.5^{t}_{o}	$22.8^{h}_{c,s}$

The free parameter values of *all* retrieval methods are set using 10-fold cross validation performed over the queries in a dataset. Query IDs are used to create the folds. The optimal parameter values for each of the 10 train sets are determined using a simple grid search applied to optimize MAP. The learned parameter values are then used for the queries in the corresponding test fold.

The value of the Dirichlet smoothing parameter, μ, is selected from $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$. The parameter λ, used in HTLM, STLM, F-HT and F-ST, is set to values in $\{0, 0.1, \ldots, 1\}$. The document (τ_d) and query (τ_q) entity-markup confidence level thresholds, used in HT, HTOEnt and F-HT, are set to values in $\{0, 0.1, \ldots, 0.9\}$.

4.2 Experimental results

4.2.1 Entity-based language models

Table 2 presents the performance of the methods that use entity-based language models to instantiate Equation 6. Our first observation is that the HT and ST methods outperform the standard term-based language-model retrieval, TermsLM, in all relevant comparisons (6 corpora × 3 evaluation measures); most improvements are substantial and statistically significant. Furthermore, HT and ST outperform to a substantial and statistically significant degree their special cases which use only entity tokens: HTOEnt and STOEnt, respectively. These findings attest to the merits of using our proposed language models, HTLM and STLM, which integrate term-based and entity-based information.

We also see in Table 2 that HT and ST outperform HTCon in most relevant comparisons; most MAP improvements for ST are statistically significant. Recall from Section 4.1 that HTCon represents past practice of concept-based representations: accounting uniformly for all entity mentions and attributing equal importance to entity and term tokens. Below we further study the importance of accounting for the confidence level of entity markups, and attributing different weights to term and entity tokens as in HT and ST.

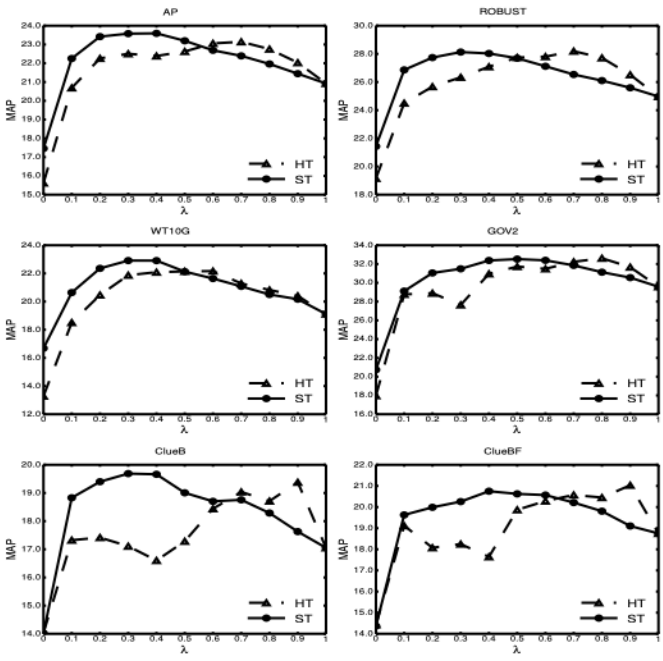

Figure 1: The effect of varying λ on the MAP of HT and ST. For $\lambda = 1$, the methods amount to TermsLM (term-based language model retrieval). For $\lambda = 0$, the methods use only entity tokens. The performance is reported for the test folds (i.e., all queries in a dataset) when fixing the value of λ and using cross validation to set the values of all other free parameters. Note: figures are not to the same scale.

Table 2 shows that ST outperforms HT in most relevant comparisons, although rarely to a statistically significant degree. In addition, ST posts more statistically significant improvements over HTCon than HT. We note that HT depends on four free parameters (λ, τ_q, τ_d and μ) while ST depends only on two (λ and μ). Furthermore, the values learned for τ_q and τ_d in HT using the training folds are very low, attesting to the merits of using high recall entity markup. (We revisit this point below.) Overall, these findings attest to the potential merits of using a soft-thresholding approach for the confidence level of entity markups (STLM) with respect to a hard-thresholding approach (HTLM); i.e., accounting for all entity markups in a text and weighing their impact by their confidence levels is superior to accounting, uniformly, for entity markups with a confidence level above a threshold.

Terms vs. entities. Figure 1 depicts the MAP performance of HT and ST as a function of λ. Low and high values of λ result in more importance attributed to entity-based and term-based information, respectively. For $\lambda = 1$, the two methods amount to TermsLM — i.e., standard term-based language model retrieval. For $\lambda = 0$, the methods use only entity-based information; specifically, HT reduces to HTOEnt and ST reduces to STOEnt.

We see in Figure 1 that optimal performance is always attained for $\lambda \notin \{0, 1\}$. This finding echoes those based on Table 2. That is, HT and ST outperform TermsLM,

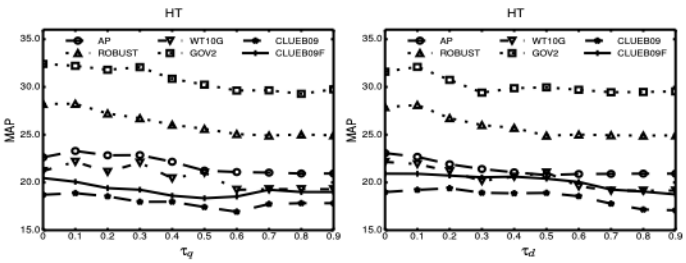

Figure 2: The effect of varying τ_q and τ_d on the MAP performance of HT. The values of free parameters, except for that in the x-axis, are set using cross validation as in Figure 1.

and HTOEnt and STOEnt, respectively. Thus, we find that there is much merit in integrating term-based and entity-based information for representing queries and documents.

Figure 1 shows that the optimal value of λ for HT is often higher than for ST. This can be attributed to the fact that HTLM, used to represent the query and documents in HT, uses a single confidence-level threshold for entity markups. Thus, potentially valuable information about entities is not utilized. As a result, HT calls for more reliance on term-based information to "compensate" for this potential information loss. In contrast, ST accounts for all entity markups, weighing their impact by their confidence levels. Hence, the "risk" in relying on entity-based information is lower[7].

To further explore the effect of using a hard threshold for the confidence level of entity markups in HT, we present in Figure 2 its MAP performance as a function of τ_q and τ_d — the query and document thresholds, respectively. Recall that low threshold corresponds to high recall markup. Figure 2 shows that low values of τ_q and τ_d lead to improved performance. This finding can be attributed to the fact that increasing the confidence-level threshold amounts to loosing potentially valuable information about appearances of entities in the query and documents. To compensate for the lower precision (i.e., noisier) markup caused by using a low threshold, more weight is put on term-based information as is evident in the relatively high optimal values of λ presented in Figure 1. Specifically, we note that the *learned values* of λ, τ_d, and τ_q, averaged over the train folds, for AP, ROBUST, WT10G, GOV2, ClueB and ClueBF are (0.6, 0.01, 0.11), (0.7,0.1,0.01), (0.55,0.1,0.2), (0.77,0.1,0.01), (0.7,0.15,0), and (0.81, 0.17, 0) respectively; namely, relatively high values of λ and low values of τ_d and τ_q lead to improved performance.

Entity linking. Our main evaluation is based on using TagMe for entity linking. In Table 3 we compare the retrieval performance when using the entity markups of TagMe and Wikifier. Having Wikifier annotate large-scale collections is a challenging computational task. Thus, we present results only for AP, ROBUST and WT10G. We report MAP and NDCG; the performance patterns for p@10 are the same.

Table 3 shows that using ST, our best performing method from above, with Wikifier, results in performance that transcends (often, significantly) that of the standard term-based language model (TermsLM) when using all queries in a dataset

[7]Setting λ on a per-query basis, in the spirit of work on fusing term-only-based and latent-entity-space-based retrieval scores [33], is a future direction we intend to explore.

Table 3: Comparing entity-linking tools. Either all queries in a dataset are used ("All Queries"), or only those marked with at least one entity by both TagMe and Wikifier ("Marked Queries"). Bold: best result in a column in a block; 't', 's', 'w' and 'e': statistically significant differences with TermsLM, TagMe-ST, Wikifier-ST and TagMe-STOEnt, respectively.

		AP		ROBUST		WT10G	
		MAP	NDCG	MAP	NDCG	MAP	NDCG
All Queries							
	TermsLM	20.9	40.4	25.0	43.5	19.1	30.3
TagMe	ST	$\mathbf{23.5}^{\mathbf{t}}$	$\mathbf{45.5}^{\mathbf{t}}$	$\mathbf{28.1}^{\mathbf{t}}$	$\mathbf{46.9}^{\mathbf{t}}$	$\mathbf{22.9}^{\mathbf{t}}$	$\mathbf{34.3}^{\mathbf{t}}$
Wikifier	ST	23.3^{t}	43.6	27.2^{t}	45.6^{t}	$19.7^{t,s}$	30.9^{s}
Marked Queries							
	TermsLM	22.2	41.7	25.4	43.9	21.4	34.2
TagMe	ST	$\mathbf{25.1}^{\mathbf{t}}$	$\mathbf{48.4}^{\mathbf{t}}$	$\mathbf{28.8}^{\mathbf{t}}$	$\mathbf{47.3}^{\mathbf{t}}$	$\mathbf{24.8}^{\mathbf{t}}$	$\mathbf{36.2}$
Wikifier	ST	$\mathbf{25.1}^{\mathbf{t}}$	46.2^{t}	28.0^{t}	46.4^{t}	21.9^{s}	34.0
TagMe	STOEnt	$18.5^{t,s}_{w}$	41.4^{s}	$22.9^{t,s}_{w}$	41.1^{s}_{w}	18.1^{s}	28.1^{s}
Wikifier	STOEnt	$17.5^{t,s}_{w}$	39.1^{s}_{w}	$19.4^{t,s}_{w,e}$	$34.8^{t,s}_{w,e}$	$12.6^{t,s}_{w,e}$	$21.8^{t,s}_{w}$

(the "All Queries" block). However, the performance of using TagMe is consistently better.

TagMe marks more queries with at least one entity than Wikifier: for AP, ROBUST and WT10G, Wikifier marked no entities in 17, 34 and 26 queries, respectively; TagMe did not mark entities in 0, 1 and 3 queries. (For GOV2 TagMe marked all queries with entities and for ClueB/ClueBF all queries except for one.) Recall that for queries with no marked entities, ST relies only on term-based information.

To refine the comparison of TagMe and Wikifier, we report the performance of ST and STOEnt[8] — the latter relies only on entity tokens — with these two tools over only queries in which both marked at least one entity. As can be seen in the "Marked Queries" block in Table 3, TagMe still outperforms Wikifier in almost all relevant comparisons; for STOEnt, several improvements are statistically significant.

TagMe's superiority can be partially attributed to marking more entities (with confidence level > 0) on average than Wikifier: (2.4, 1.8, 2.0) with respect to (1.7, 1.2, 1.0) in queries over AP, ROBUST and WT10G; and, (157.2, 158.7, 207.0) with respect to (58.4, 50.5, 61.7) in documents.

To conclude, our methods are effective with both TagMe and Wikifier. Using TagMe yields better performance that can be partially attributed to higher recall entity markup.

4.2.2 *The score-based fusion methods*

Table 4 presents the performance of the F-HT and F-ST methods from Section 3.2.1 that perform score fusion of term-only-based and entity-only-based retrieval scores. The performance of TermsLM (term-only language model), HT and ST that integrate term and entity information at the language model level, and that of F-HTCon which is a special case of F-HT (see Section 4.1), is presented for reference. We see that F-HT and F-ST substantially outperform TermsLM. (F-ST posts the best performance in most relevant comparisons in Table 4.) Both methods also outperform F-HTCon in most relevant comparisons.

[8]For queries for which a tool does not mark any entities, no documents are retrieved with STOEnt. Thus, we do not report the performance of STOEnt using all queries as the results are inherently biased in favor of TagMe which marks many more queries with entities than Wikifier.

Table 4: Score-based fusion ("F-" methods). Bold: best result in a row; 't', 'h', 's', 'f' and 'c': statistically significant differences with TermsLM, HT, ST, F-HT and F-HTCon, respectively.

		TermsLM	HT	ST	F-HT	F-HTCon	F-ST
AP	MAP	20.9	23.1^{t}	23.5^{t}	23.1^{t}	22.5_{s}	$\mathbf{23.9^{t,h}_{f,c}}$
	p@10	39.1	44.2^{t}	43.8^{t}	$\mathbf{44.5^{t}}$	43.5^{t}	44.2^{t}
	NDCG	40.4	45.3^{t}	45.5^{t}	$\mathbf{46.2^{t}}$	45.1^{t}	45.8^{t}
ROBUST	MAP	25.0	28.1^{t}	28.1^{t}	28.1^{t}	27.7^{t}	$\mathbf{28.4^{t}_{c}}$
	p@10	42.2	45.5^{t}	45.3^{t}	45.7^{t}	45.2^{t}	$\mathbf{46.7^{t}_{s,c}}$
	NDCG	43.5	47.1^{t}	46.9^{t}	47.3^{t}	46.6^{t}	$\mathbf{47.8^{t}_{c}}$
WT10G	MAP	19.1	21.9^{t}	$\mathbf{22.9^{t,h}}$	22.2^{t}	21.6^{t}_{s}	$\mathbf{22.9^{t}_{c}}$
	p@10	27.3	30.4^{t}	31.6^{t}	30.0	30.4^{t}	$\mathbf{31.8^{t}}$
	NDCG	30.3	32.7	$\mathbf{34.3^{t}}$	32.7	33.1	33.7^{t}
GOV2	MAP	29.6	32.1^{t}	32.2^{t}	$\mathbf{33.5^{t,h}_{s}}$	$30.6^{h}_{s,f}$	$33.3^{t,h}_{s,c}$
	p@10	53.9	57.3^{t}	57.7^{t}	$\mathbf{58.6^{t}}$	57.0	58.0^{t}
	NDCG	44.8	47.4^{t}	47.9^{t}	$\mathbf{48.7^{t}}$	46.6	48.2^{t}
ClueB	MAP	17.1	18.7^{t}	19.5^{t}	$19.6^{t,h}$	19.3^{t}	$\mathbf{20.8^{t,h}_{s,f,c}}$
	p@10	22.7	25.9^{t}	27.4^{t}	26.4^{t}	27.5^{t}	$\mathbf{28.8^{t,h}_{f}}$
	NDCG	16.5	18.7^{t}	19.3^{t}	19.1^{t}	19.9^{t}	$\mathbf{20.5^{t,h}_{f}}$
ClueBF	MAP	18.8	20.5^{t}	20.3^{t}	$21.3^{t,h}$	19.7_{f}	$\mathbf{21.8^{t,h}_{s,c}}$
	p@10	33.6	37.9^{t}	37.9^{t}	$\mathbf{39.6^{t}}$	36.5_{f}	39.4^{t}_{c}
	NDCG	24.3	28.4^{t}	27.5^{t}	$\mathbf{29.5^{t}_{s}}$	27.6	29.2^{t}_{s}

Table 5: Comparison and integration with SDM [36]. Bold: the best result in a row. 't', 's', 'f' and 'm' mark statistically significant differences with TermsLM, ST, F-ST and SDM, respectively.

		TermsLM	ST	F-ST	SDM	SDM+STLM
AP	MAP	20.9	23.5^{t}	$\mathbf{23.9^{t}}$	21.6^{s}_{f}	$\mathbf{23.9^{t}_{m}}$
	p@10	39.1	43.8^{t}	$\mathbf{44.2^{t}}$	40.6_{f}	$\mathbf{44.2^{t}_{m}}$
	NDCG	40.4	45.5^{t}	$\mathbf{45.8^{t}}$	42.3_{f}	$\mathbf{45.8^{t}_{m}}$
ROBUST	MAP	25.0	28.1^{t}	$\mathbf{28.4^{t}}$	$25.7^{t,s}_{f}$	28.3^{t}_{m}
	p@10	42.2	45.3^{t}	$\mathbf{46.7^{t,s}}$	43.9^{t}_{f}	$45.7^{t}_{f,m}$
	NDCG	43.5	46.9^{t}	$\mathbf{47.8^{t}}$	$44.8^{t,s}_{f}$	$47.1^{t}_{f,m}$
WT10G	MAP	19.1	22.9^{t}	22.9^{t}	20.2^{s}_{f}	$\mathbf{23.1^{t}_{m}}$
	p@10	27.3	31.6^{t}	$\mathbf{31.8^{t}}$	27.7^{s}_{f}	31.6^{t}_{m}
	NDCG	30.3	$\mathbf{34.3^{t}}$	33.7^{t}	30.7^{s}_{f}	34.0^{t}_{m}
GOV2	MAP	29.6	32.2^{t}	$33.3^{t,s}$	32.1^{t}	$\mathbf{34.7^{t,s}_{f,m}}$
	p@10	53.9	57.7^{t}	58.0^{t}	58.3^{t}	$\mathbf{61.4^{t,s}_{f,m}}$
	NDCG	44.8	47.9^{t}	48.2^{t}	48.4^{t}	$\mathbf{50.6^{t,s}_{f,m}}$
ClueB	MAP	17.1	19.5^{t}	$20.8^{t,s}$	$18.2^{t,s}_{f}$	$\mathbf{21.5^{t,s}_{m}}$
	p@10	22.7	27.4^{t}	28.8^{t}	23.8^{s}_{f}	$\mathbf{30.8^{t,s}_{f,m}}$
	NDCG	16.5	19.3^{t}	20.5^{t}	16.9^{s}_{f}	$\mathbf{21.9^{t,s}_{m}}$
ClueBF	MAP	18.8	20.3^{t}	$21.8^{t,s}$	20.2^{t}_{f}	$\mathbf{22.7^{t,s}_{f,m}}$
	p@10	33.6	37.9^{t}	39.4^{t}	35.8^{t}_{f}	$\mathbf{42.8^{t,s}_{f,m}}$
	NDCG	24.3	27.5^{t}	$29.2^{t,s}$	25.9^{t}_{f}	$\mathbf{32.2^{t,s}_{f,m}}$

In most relevant comparisons, F-HT outperforms HT and F-ST outperforms ST, although most performance differences are not statistically significant. The improvements can be attributed to the fact that F-HT and F-ST use a different smoothing parameter value for terms and entities while HT and ST use a joint one. (See Section 3.2.1 for details.)

The potential effectiveness of using different smoothing parameters for term and entity tokens stems from the different number of terms and entity markups in a document. The average number of terms in a document for AP, ROBUST, WT10G, GOV2, and ClueB (ClueBF) is 455.4, 474.8, 588.2, 904.7 and 813.6, respectively. The average number of entity markups with a confidence level > 0 is much lower: 157.2, 158.7, 207.0, 291.9 and 307.8.

4.2.3 Comparison and integration with SDM

We next compare our entity-based approach with the sequential dependence model (SDM) [36] which scores d by:

$$S_{SDM}(d;q) \stackrel{def}{=} \lambda_S Sim_S(d,q) + \lambda_O Sim_O(d,q) + \lambda_U Sim_U(d,q);$$

the sum of the λ_S, λ_O and λ_U parameters is 1; $Sim_S(d,q)$, $Sim_O(d,q)$ and $Sim_U(d,q)$ are cross-entropy based similarity estimates of the document to the query, utilizing information about occurrences of unigram, ordered bigrams, and unordered bigrams, respectively, of q's terms in d; un-ordered bigrams are confined to 8-terms windows in documents.

Using entity tokens in our methods amounts to utilizing information about the occurrences of only *some ordered* variable-length n-grams of query terms in documents — i.e., n-grams which constitute entities. Thus, in contrast to SDM, our methods do not utilize proximity information for query terms which are not in entity markups nor proximity information for unordered n-grams of query terms.

In addition, we study the merit of integrating entity-based information, specifically, our soft-thresholding language model STLM, with SDM. To that end, we augment the SDM scoring function with an entity-based document-query similarity estimate, $Sim_E(d,q)$. For this estimate, we use the score assigned to d by the STOEnt method; i.e., we use an entity-only language model since term-based information is accounted for in $Sim_S(d,q)$. The resultant method, **SDM+STLM**, scores d by ($\lambda_S + \lambda_O + \lambda_U + \lambda_E = 1$):

$$S_{SDM+STLM}(d;q) \stackrel{def}{=} \lambda_S Sim_S(d,q) + \lambda_O Sim_O(d,q) + \lambda_U Sim_U(d,q) + \lambda_E Sim_E(d,q).$$

SDM+STLM can be viewed as a novel instantiation of a weighted dependence model (WSDM) [4] with a novel concept type (i.e., entity). If $\lambda_O = \lambda_U = 0$, SDM+STLM amounts to our F-ST method (see Section 3.2.1).

All free parameters of SDM and SDM+STLM: λ_S, λ_O, λ_U, λ_E and the Dirichlet smoothing parameter, μ, are set using cross validation as described in Section 4.1; λ_S, λ_O, λ_U, and λ_E are selected from $\{0, 0.1, \ldots, 1\}$ and μ is set to values in $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$.

Table 5 shows that ST and F-ST outperform SDM, often statistically significantly, in most relevant comparisons (6 corpora × 3 evaluation measures). This implies that using variable length n-grams which potentially bear semantic meaning (entities) can yield better performance than using ordered and unordered bigrams which do not necessarily have semantic meaning. Recall that in contrast to SDM, ST and F-ST do not account for proximities between terms which do not constitute entities and for unordered bigrams.

In most relevant comparisons, SDM+STLM outperforms SDM and ST (which utilizes STLM) and is as effective as, and often posts statistically significant improvements over, F-ST — its special case that fuses unigram term-only and entity-only retrieval scores. The few cases where F-ST outperforms SDM+STLM could be attributed to potential overfitting effects due to the high number of free parameters of SDM+STLM and the relatively low number of queries.

We also found that effective weights assigned to entity-only similarities in SDM+STLM (λ_E) are much higher than those assigned to ordered (λ_O) and un-ordered (λ_U) bigram

Table 6: Robustness analysis. Number of queries for which ST hurts (-) and improves (+) AP performance with respect to TermsLM and SDM.

	AP		ROBUST		WT10G		GOV2		ClueB		ClueBF	
	-	+	-	+	-	+	-	+	-	+	-	+
ST vs. TermsLM	38	61	75	173	31	63	50	99	54	137	75	112
ST vs. SDM	35	64	87	161	33	60	74	75	79	112	89	97

term-based similarities. Furthermore, effective values of λ_O and λ_U are lower and higher, respectively, for SDM+STLM than for SDM. These findings further attest to the merits of using entity-based similarities with respect to (ordered and un-ordered) bigram similarities, and show that un-ordered bigram, in contrast to ordered bigram, similarities could be complementary to entity-based similarities.

4.2.4 Further analysis

We now turn to further analyze merits, and shortcomings, of using entity-based query and document representations. To that end, we focus on the ST method that utilizes STLM.

Table 6 presents performance robustness analysis: the number of queries for which ST improves or hurts average precision (AP) over TermsLM and SDM. In both cases, ST improves AP for more queries than it hurts; naturally, the differences with SDM are smaller than those with TermsLM.

One advantage of STLM is that it represents the query and documents using entities which constitute variable length n-grams with semantic meaning. A case in point, query #41 in ClueWeb, "orange county convention center", refers to the primary public convention center for the Central Florida region. TermsLM, SDM and ST ranked the Web home page for this entity second. However, at the third rank in the lists retrieved by TermsLM and SDM appears a Wikipedia page titled "list of convention and exhibition centers", which is not specific to the entity of concern. The average precision (AP) of TermsLM, SDM and ST for the query in the ClueB dataset was 9, 13, and 30, respectively, attesting to the merit of the correct identification of the entity in the query and its utilization by ST.

The ST method can suffer from incorrect entity identification in queries. For example, query #407 in ROBUST, "poaching, wildlife preserves", targets information about the impact of poaching on the world's various wildlife preserves. The entities identified by TagMe are "poaching", "wildlife" and "preserves"; the latter refers to fruit preserves instead of nature preserves. Such erroneous entity identification can be attributed to the little context short queries provide. Consequently, the AP of ST for this query is only 8 while that of TermsLM and SDM is 31.4 and 30.0, respectively.

4.3 Using entity-based language models in additional retrieval paradigms

We next explore the effectiveness of using our entity-based language models in two additional retrieval paradigms: cluster-based document retrieval and query expansion.

4.3.1 Cluster-based document retrieval

Let D_{init} denote the list of top-n documents retrieved by TermsLM (standard language-model-based retrieval). Following common practice in work on cluster-based document retrieval [32, 24], we re-rank D_{init} using information induced from nearest-neighbor clusters of documents in D_{init}.

Table 7: Cluster-based document re-ranking. Bold: the best result in a row; 't', 's', '$*$' and 'ψ' mark statistically significant differences with TermsLM, ST, C-Term-Term and C-Term-Ent, respectively.

		TermsLM	ST	C-Term-Term	C-Term-Ent	C-Ent-Ent
AP	p@10	39.6	42.5	43.2^{t}	44.3^{t}	$\mathbf{46.5}^{\mathbf{t,s}}$
	NDCG	40.8	44.8^{t}	44.2^{t}	44.9	$\mathbf{46.8}^{\mathbf{t}}$
ROBUST	p@10	42.2	44.3^{t}	43.1	46.0^{t}_{*}	$\mathbf{47.7}^{\mathbf{t,s}}_{*,\psi}$
	NDCG	43.5	45.5^{t}	44.2	47.5^{t}_{*}	$\mathbf{49.1}^{\mathbf{t,s}}_{*,\psi}$
WT10G	p@10	28.6	30.6	30.2	$33.7^{t,s}_{*}$	$\mathbf{34.8}^{\mathbf{t,s}}_{*}$
	NDCG	31.2	33.4	32.1	35.4^{t}_{*}	$\mathbf{36.3}^{\mathbf{t,s}}_{*}$
GOV2	p@10	53.4	57.0^{t}	55.1	$\mathbf{58.3}^{\mathbf{t}}$	57.9^{t}
	NDCG	45.0	46.8	45.8	$\mathbf{48.9}^{\mathbf{t}}$	47.8^{t}
ClueB	p@10	23.7	27.1^{t}	23.7	$\mathbf{33.0}^{\mathbf{t,s}}_{*}$	$31.5^{t,s}_{*}$
	NDCG	17.2	19.1	17.2	$\mathbf{24.9}^{\mathbf{t,s}}_{*}$	$22.9^{t,s}_{*}$
ClueBF	p@10	32.1	36.9^{t}	31.2^{s}	38.5^{t}_{*}	$\mathbf{39.0}^{\mathbf{t}}_{*}$
	NDCG	22.9	27.8^{t}	23.1^{s}	$\mathbf{30.3}^{\mathbf{t}}_{*}$	29.6^{t}_{*}

We use $Sim(x,y) \stackrel{def}{=} exp(-CE\left(\theta_x^{MLE} \mid\mid \theta_y^{Dir}\right))$ to measure the similarity between texts x and y [24]; θ_x^{MLE} is an unsmoothed MLE induced from x and θ_y^{Dir} is a Dirichlet smoothed language model induced from y. Each document d ($\in D_{init}$) and the $k-1$ documents d' ($d' \neq d$) in D_{init} that yield the highest $Sim(d,d')$ constitute a cluster.

We rank the (overlapping) clusters c, each contains k documents, by: $\sqrt[k]{\prod_{d \in c} Sim(q,d)}$ [32]. This is a highly effective simple cluster ranking method [24]. To induce document ranking, each cluster is replaced with its constituent documents omitting repeats; documents in a cluster are ordered by their query similarity: $Sim(q,d)$.

The document (re-)ranking procedure just described relies on the choice of the document language models used to induce clusters (i.e., in $Sim(d,d')$) and the choice of document and query language models used to induce document-query similarities ($Sim(q,d)$); the latter are used for ranking both clusters and documents within the clusters. We use **C-Term-Term** to denote the standard method that uses term-only language models for inducing clusters and document-query similarities [32, 24]. The **C-Term-Ent** method utilizes the same clusters used by C-Term-Term, but uses our entity-based language model, STLM, for inducing document-query similarities to rank clusters and documents in them. In the **C-Ent-Ent** method, STLM is used to both create clusters and induce document-query similarities. As a reference comparison, we re-rank D_{init} using the ST method that uses STLM but does not utilize clusters.

As the main goal of cluster-based re-ranking is improving precision at top ranks [32, 24], we report p@10 and NDCG@10 (NDCG). Free-parameter values are set using cross validation; NDCG is the optimization criterion. Specifically, n is selected from $\{50, 100\}$; k is in $\{5, 10\}$; and, λ (used in STLM) is in $\{0, 0.1, \ldots, 1\}$; the Dirichlet smoothing parameter is set to 1000. Table 7 presents the results.

We see that all cluster-based methods (denoted "C-X-Y") almost always outperform the initial term-based document ranking, TermsLM. C-Term-Ent substantially outperforms C-Term-Term. This attests to the merits of using STLM for inducing cluster ranking and within cluster document ranking. In most relevant comparisons, C-Ent-Ent outperforms (and is never statistically significantly outperformed by) C-Term-Ent, attesting to the potential merits of using

Table 8: Query expansion. Bold: the best result in a row. 't', 's', 'r', 'w', 'm' and 'n' mark statistically significant differences with TermsLM, ST, RM3, WikiRM, SDM-RM and RMST, respectively.

		TermsLM	ST	RM3	WikiRM	SDM-RM	RMST	RMST-ST
AP	MAP	20.9	23.5^{t}	24.1^{t}	24.0^{t}	24.9^{t}	24.6^{t}	$\mathbf{27.4}^{\mathbf{t,s,r}}_{\mathbf{w,m,n}}$
	p@10	39.1	43.8^{t}	42.5^{t}	46.2^{t}	43.9^{t}	44.8^{t}	$\mathbf{46.8}^{\mathbf{t,r}}$
	NDCG	40.4	45.5^{t}	43.2	$\mathbf{48.2}^{\mathbf{t,r}}$	45.6^{t}	45.0^{t}	$47.4^{t,r}$
ROBUST	MAP	25.0	28.1^{t}	28.3^{t}	27.8^{t}	28.4^{t}	29.0^{t}	$\mathbf{30.5}^{\mathbf{t,s,r}}_{\mathbf{w,m,n}}$
	p@10	42.2	45.3^{t}	43.6	44.6^{t}	43.2	$45.9^{t,r}_{m}$	$\mathbf{47.1}^{\mathbf{t,s,r}}_{\mathbf{w,m}}$
	NDCG	43.5	46.9^{t}	43.8^{s}	$46.1^{t,r}$	43.6^{s}_{w}	$46.5^{t,r}_{m}$	$\mathbf{47.2}^{\mathbf{t,r}}_{\mathbf{m}}$
WT10G	MAP	19.1	$\mathbf{22.9}^{\mathbf{t}}$	19.6^{s}	$21.9^{t,r}$	20.0^{s}	$22.7^{t,r}_{m}$	$22.8^{t,r}_{m}$
	p@10	27.3	31.6^{t}	28.0^{s}	$\mathbf{34.2}^{\mathbf{t,r}}$	28.6_{w}	$31.7^{t,r}_{m}$	31.1^{t}
	NDCG	30.3	$\mathbf{34.3}^{\mathbf{t}}$	30.1^{s}	$\mathbf{34.3}^{\mathbf{t,r}}$	30.5^{s}_{w}	32.9	31.8^{s}
GOV2	MAP	29.6	32.2^{t}	32.4^{t}	32.1^{t}	$\mathbf{33.7}^{\mathbf{t}}_{\mathbf{w}}$	33.1^{t}	$\mathbf{33.7}^{\mathbf{t,s}}$
	p@10	53.9	57.7^{t}	58.1^{t}	$\mathbf{60.1}^{\mathbf{t}}$	58.0^{t}	59.6^{t}	58.5^{t}
	NDCG	44.8	47.9^{t}	48.0^{t}	$\mathbf{50.6}^{\mathbf{t}}$	47.6	49.4^{t}	48.8^{t}
ClueB	MAP	17.1	19.5^{t}	19.3^{t}	$21.9^{t,s,r}$	$20.9^{t,r}$	$20.7^{t,s,r}$	$\mathbf{22.1}^{\mathbf{t,s,r}}_{\mathbf{n}}$
	p@10	22.7	27.4^{t}	30.6^{t}	$\mathbf{35.3}^{\mathbf{t,s,r}}$	$32.2^{t,s}_{w}$	$32.2^{t,s}_{w}$	$34.9^{t,s,r}_{n}$
	NDCG	16.5	19.3^{t}	$22.6^{t,s}$	$26.1^{t,s,r}$	$24.3^{t,s}$	$25.1^{t,s,r}$	$\mathbf{27.1}^{\mathbf{t,s,r}}_{\mathbf{n}}$
Clue BF	MAP	18.8	20.3^{t}	20.4^{t}	21.0^{t}	$21.8^{t,s,r}$	20.8^{t}	$\mathbf{21.9}^{\mathbf{t,s}}_{\mathbf{n}}$
	p@10	33.6	37.9^{t}	37.9^{t}	38.5^{t}	$\mathbf{39.7}^{\mathbf{t,r}}$	38.2^{t}	38.4^{t}
	NDCG	24.3	27.5^{t}	28.1^{t}	28.2^{t}	$29.8^{t,r}$	28.5^{t}	$\mathbf{30.3}^{\mathbf{t,s}}$

entity-based information to also create clusters. However, only two improvements are statistically significant.

Finally, Table 7 shows that in almost all relevant comparisons, ST outperforms TermsLM (often, statistically significantly) and C-Term-Term and is outperformed by C-Term-Ent and C-Ent-Ent. This shows that while there is merit in using STLM for direct ranking of documents as shown in Section 4.2.1, the performance can be further improved by using STLM for cluster-based document ranking.

4.3.2 *Query expansion*

As noted in Section 2, there is much work on expanding queries with terms and entities using entity-based information. In contrast, our entity-based language models, when induced from the query, utilize only query terms and entities marked in the query. Hence, we study the effectiveness of using our language models to perform query expansion.

We use the relevance model (RM3) [1] as a basis for instantiating expanded query forms. The probability assigned to *token* t by a relevance model RM is:

$$RM(t) \stackrel{def}{=} \alpha\theta_q^{MLE}(t) + (1-\alpha)\sum_{d\in L}\theta_d^{Dir}(t)\frac{S(d;q)}{\sum_{d'\in L}S(d';q)}; \quad (8)$$

α is a free parameter; L is a list of top-retrieved documents used to construct RM; $S(d;q)$ is d's score. Due to computational considerations, as in work on entity-based query expansion [13, 45] we use RM to re-rank an initially retrieved document list; $CE\left(RM \mid\mid \theta_d^{Dir}\right)$ serves for re-ranking.

Using only terms as tokens, and applying standard language-model-based retrieval (TermsLM) over the corpus to create L, yields the standard **RM3** [1]. Creating L by applying TermsLM over Wikipedia results in **WikiRM** [46], an external corpus expansion approach also used in [13, 45]. RM3 and WikiRM re-rank a document list retrieved by TermsLM. (WikiRM is the only model where the list from which RM is constructed, L, is not a sub-set of the list to be re-ranked.) In both methods, $S(d;q) \stackrel{def}{=} \exp(-CE\left(\theta_q^{MLE} \mid\mid \theta_d^{Dir}\right))$.

The **SDM-RM** model [13] is constructed from, and used to re-rank, lists retrieved by the sequential dependence model (SDM) [36]. θ_d^{Dir}, and the resultant relevance model constructed by setting $\alpha = 0$ in Equation 8, are term-based unigram language models; $S(d;q)$ is the exponent of the score assigned to d by SDM. Re-ranking is performed by linear interpolation of the SDM score assigned to d and $CE\left(RM \mid\mid \theta_d^{Dir}\right)$, using a parameter α. SDM-RM is, in fact, the highly effective Latent Concept Expansion method [37] without IDF-based weighting of expansion terms.

The next two relevance models, defined over $\mathcal{T}$ (the term-entity token space from Equation 1), are novel to this study. They utilize our STLM language model which integrates terms and entities at the language model level. **RMST** is inspired by methods proposed by Dalton et al. [13][9] by the virtue of using both terms and entities for query expansion. θ_q^{MLE} and θ_d^{Dir} are our STLM language models. $S(d;q) \stackrel{def}{=} \exp(-CE\left(\theta_q^{MLE} \mid\mid \theta_d^{Dir}\right))$. The TermsLM method is applied over the corpus to create the initial list to be re-ranked (cf. [45]) and from which L is derived.

RMST-ST is constructed as RMST using STLM. The difference is that our entity-based ST method, rather than TermsLM, is used to create the initial list to be re-ranked and from which L is derived. The formal ease of using STLM in the relevance model (Equation 8), yielding RMST and RMST-ST, attests to the merits of using a single language model defined over terms and entities with respect to the alternative score-based fusion approach from Section 3.2.1.

The free parameters of all methods are set using cross validation. The number of expansion terms (i.e., those assigned the highest probability by RM), the number of documents in L, and α are set to values in $\{10, 30, 50, 100\}$, $\{50, 100\}$ and $\{0, 0.1, \ldots, 1\}$, respectively. (Only for WikiRM, the number of documents in L is selected from $\{1, 5, 10, 30, 50, 100\}$ following [46].) All lists that are re-ranked contain 1000 documents. The values of the free parameters of ST and SDM are selected from the ranges specified in Section 4.1. The Dirichlet smoothing parameter, μ, is selected from $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$; for relevance model construction (Equation 8) the value 0 is also used (yielding unsmoothed MLE). To reduce the number of free-parameter values configurations, we use the same value of μ for creating L, for re-ranking and for constructing the relevance model, unless 0 is used for relevance model construction.

Table 8 presents the performance. Our ST method, which does not perform query expansion, is competitive with the term-based relevance model (RM3). We also see that RMST is an effective expansion method which often outperforms RM3 and SDM-RM. This finding echoes those from past work [13, 45] about the merits of using both terms and entities for query expansion. The best performing method in most relevant comparisons is RMST-ST which uses STLM to (i) create an effective initial list for re-ranking; (ii) create an effective list, L, for relevance model construction; and, (iii) induce ranking using the entity-based relevance model as in RMST. We conclude that our STLM language model can play different important roles in query expansion.

Table 8 shows that expansion using Wikipedia as an external corpus (WikiRM) is effective. Our RMST and RMST-ST expansion methods (as well as ST) utilize entity tokens marked by TagMe (i.e., Wikipedia concepts), but do no use

[9]Various expansion methods, which utilize also auxiliary information about entities from the entity repository, were integrated in [13]. We do not use such auxiliary information.

the text on their Wikipedia pages in contrast to WikiRM. Thus, integrating WikiRM with our methods, e.g., using score-based integration [13], is interesting future direction.

5. CONCLUSIONS

We presented novel entity-based language models induced using an entity linking tool. The models simultaneously account for the uncertainty in the entity-linking process and the balance between using term-based and entity-based information. We showed the merits of using the language models for document retrieval in several retrieval paradigms.

Acknowledgments. We thank the reviewers for their comments. This paper is based upon work supported in part by a Yahoo! faculty research and engagement award.

6. REFERENCES

[1] N. Abdul-jaleel, J. Allan, W. B. Croft, O. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at TREC 2004: Novelty and hard. In *Proc. of TREC-13*, 2004.

[2] J. Allan, J. P. Callan, W. B. Croft, L. Ballesteros, J. Broglio, J. Xu, and H. Shu. Inquery at TREC-5. In *Proc. of TREC-5*, pages 119–132, 1996.

[3] A. R. Aronson, T. C. Rindflesch, and A. C. Browne. Exploiting a large thesaurus for information retrieval. In *Proc. of RIAO*, volume 94, pages 197–216, 1994.

[4] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proc. of WSDM*, pages 31–40, 2010.

[5] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proc. of SIGIR*, pages 605–614, 2011.

[6] M. Bendersky, D. Metzler, and W. B. Croft. Effective query formulation with multiple information sources. In *Proc. of WSDM*, pages 443–452, 2012.

[7] W. C. Brandão, R. L. T. Santos, N. Ziviani, E. S. de Moura, and A. S. da Silva. Learning to expand queries using entities. *JASIST*, 65(9):1870–1883, 2014.

[8] G. Cao, J. Nie, and J. Bai. Integrating word relationships into language models. In *Proc. of SIGIR*, pages 298–305, 2005.

[9] X. Cheng and D. Roth. Relational inference for wikification. In *Proc. of EMNLP*, pages 1787–1796, 2013.

[10] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proc. of CIKM*, pages 704–711, 2005.

[11] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[12] M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proc. of WWW*, pages 249–260, 2013.

[13] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proc. of SIGIR*, pages 365–374, 2014.

[14] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.

[15] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proc. of CIKM*, pages 1625–1628, 2010.

[16] W. R. Hersh, D. H. Hickam, and T. Leone. Words, concepts, or both: optimal indexing units for automated information retrieval. In *Proc. of SCAMC*, page 644, 1992.

[17] D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proc. of SIGIR*, pages 35–41, 2002.

[18] M. Hsu, M. Tsai, and H. Chen. Combining wordnet and conceptnet for automatic query expansion: A learning approach. In *Proc. of AIRS*, pages 213–224, 2008.

[19] S. Huston and W. B. Croft. A comparison of retrieval models using term dependencies. In *Proc. of CIKM*, pages 111–120, 2014.

[20] A. Kotov and C. Zhai. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In *Proc. of WSDM*, pages 403–412, 2012.

[21] R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, 10(2):115–141, 1992.

[22] G. Kumaran and J. Allan. A case for shorter queries, and helping users create them. In *Proc. of NAACL*, pages 220–227, 2007.

[23] H.-K. J. Kuo and W. Reichl. Phrase-based language models for speech recognition. In *Proc. of EUROSPEECH*, 1999.

[24] O. Kurland and E. Krikon. The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research (JAIR)*, 41:367–395, 2011.

[25] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[26] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[27] M. Levit, S. Parthasarathy, S. Chang, A. Stolcke, and B. Dumoulin. Word-phrase-entity language models: getting more mileage out of n-grams. In *Proc. of INTERSPEECH*, pages 666–670, 2014.

[28] H. Li and J. Xu. Semantic matching in search. *Foundations and Trends in Information Retrieval*, 7(5):343–469, 2014.

[29] R. Li, L. Hao, P. Zhang, D. Song, and Y. Hou. A query expansion approach using entity distribution based on markov random fields. In *Proc. of AIRS*, 2015.

[30] S. Liu, F. Liu, C. T. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proc. of SIGIR*, pages 266–272, 2004.

[31] X. Liu, F. Chen, H. Fang, and M. Wang. Exploiting entity relationship for query expansion in enterprise search. *Information Retrieval Journal*, 17(3):265–294, 2014.

[32] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proc. of ECIR*, pages 454–462, 2008.

[33] X. Liu and H. Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, December 2015.

[34] R. Mandala, T. Tokunaga, and H. Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. In *Proc. of SIGIR*, pages 191–197, 1999.

[35] E. Meij, D. Trieschnigg, M. de Rijke, and W. Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing & Management*, 46(4):448–469, 2010.

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

[37] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *Proc. of SIGIR*, pages 311–318, 2007.

[38] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. of CIKM*, pages 509–518, 2008.

[39] D. Pan, P. Zhang, J. Li, D. Song, J. Wen, Y. Hou, B. Hu, Y. Jia, and A. N. D. Roeck. Using Dempster-Shafer's evidence theory for query expansion based on freebase knowledge. In *Proc. of AIRS*, pages 121–132, 2013.

[40] C. Shah and W. B. Croft. Evaluating high accuracy retrieval techniques. In *Proc. of SIGIR*, pages 2–9, 2004.

[41] P. Srinivasan. Query expansion and medline. *Information Processing & Management*, 32(4):431–443, 1996.

[42] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proc. of SIGIR*, pages 171–180, 1993.

[43] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proc. of SIGIR*, pages 61–69, 1994.

[44] C. Xiong and J. Callan. EsdRank: Connecting query and documents through external semi-structured data. In *Proc. of CIKM*, pages 951–960, 2015.

[45] C. Xiong and J. Callan. Query expansion with Freebase. In *Proc. of ICTIR*, pages 111–120, 2015.

[46] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on Wikipedia. In *Proc. of SIGIR*, pages 59–66, 2009.

[47] Y. Yang and C. G. Chute. Words or concepts: the features of indexing units and their optimal use in information retrieval. In *Proc. of SCAMC*, page 685, 1993.

[48] C. Zhai. Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

[49] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.