Combining Term-Based and Category-Based Representations for Entity Search

Krisztian Balog, Marc Bron, Maarten de Rijke, and Wouter Weerkamp

ISLA, University of Amsterdam, Science Park 107, 1098 XG Amsterdam, The Netherlands
{k.balog,m.bron,derijke,w.weerkamp}@uva.nl

Abstract. We describe our participation in the INEX 2009 Entity Ranking track. We employ a probabilistic retrieval model for entity search in which term-based and category-based representations of queries and entities are effectively integrated. We demonstrate that our approach achieves state-of-the-art performance on both the entity ranking and list completion tasks.

1 Introduction

Our aim for the INEX 2009 Entity Ranking track was to evaluate a recently proposed probabilistic framework for entity retrieval that explicitly models category information in a theoretically transparent manner [2]. Information needs and entities are represented as tuples: each consists of a term-based model plus a category-based model, both characterized by probability distribution over words. Ranking of entities is then based on similarity to the query, measured in terms of similarity between probability distributions. In our participation, our focus is on two core steps: (1) entity modeling and (2) query modeling and expansion. Moreover, we seek to answer how well parameter settings trained on the 2007 and 2008 editions of the Entity Ranking track perform on the 2009 setup. We find that our probabilistic approach is indeed very effective on the 2009 edition of the tasks, and delivers top performance on both tasks.

The remainder of this paper is organized as follows. We introduce our retrieval model in Section 2. Next, we discuss the submitted runs (Section 3), followed by their results and a discussion of these results (Section 4). We conclude in Section 5.

2 Modeling Entity Ranking

In this section we present a probabilistic retrieval framework for the two tasks that have been formulated within the Entity Ranking track. In the *entity ranking* task we are given a query (Q) and a set of target categories (C) and have to return entities. For *list completion* we need to return entities given a query (Q), a set of similar entities (E), and (optionally also) a set of target categories (C).[1]

[2] recently proposed a probabilistic retrieval model for entity search, in which term-based and category-based representations of queries and entities are effectively

[1] We use q to denote the information need provided by the user (i.e., all components), and Q for the textual part of the query.

S. Geva, J. Kamps, and A. Trotman (Eds.): INEX 2009, LNCS 6203, pp. 265–272, 2010.
© Springer-Verlag Berlin Heidelberg 2010

integrated. With the exception of the formula used for weighting terms for query expansion, we present the original approach unchanged.

The remainder of this section is organized as follows. In §2.1 we introduce our general scheme for ranking entities. This is followed by a discussion of the two main components of our framework: entity and query models in §2.2 and §2.3, respectively.

2.1 Modeling Entity Ranking

We rank entities e according to their probability of being relevant given the query q: $P(e|q)$. Instead of estimating this probability directly, we apply Bayes' rule and rewrite it to:

$$P(e|q) \propto P(q|e) \cdot P(e), \tag{1}$$

where $P(q|e)$ expresses the probability that query q is generated by entity e, and $P(e)$ is the *a priori* probability of e being relevant, i.e., the entity prior.

Each entity is represented as a pair: $\theta_e = (\theta_e^T, \theta_e^C)$, where θ_e^T is a distribution over terms and θ_e^C is a distribution over categories. Similarly, the query is also represented as a pair: $\theta_q = (\theta_q^T, \theta_q^C)$, which is then (optionally) refined further, resulting in an expanded query model that is used for ranking entities.

The probability of an entity generating the query is estimated using a mixture model:

$$P(q|e) = \lambda \cdot P(\theta_q^T|\theta_e^T) + (1-\lambda) \cdot P(\theta_q^C|\theta_e^C), \tag{2}$$

where λ controls the interpolation between the term-based and category-representations. The estimation of $P(\theta_q^T|\theta_e^T)$ and $P(\theta_q^C|\theta_e^C)$ requires a measure of the difference between two probability distributions. We opt for the Kullback-Leibler divergence—also known as the relative entropy. The term-based similarity is estimated as follows:

$$P(\theta_q^T|\theta_e^T) \propto -KL(\theta_q^T||\theta_e^T) = -\sum_t P(t|\theta_q^T) \cdot \frac{P(t|\theta_q^T)}{P(t|\theta_e^T)}, \tag{3}$$

where the probability of a term given an entity model ($P(t|\theta_e^T)$) and the probability of a term given the query model ($P(t|\theta_q^T)$) remain to be defined. Similarly, the category-based component of the mixture in Eq. 2 is calculated as:

$$P(\theta_q^C|\theta_e^C) \propto -KL(\theta_q^C||\theta_e^C) = -\sum_c P(c|\theta_q^C) \cdot \frac{P(c|\theta_q^C)}{P(c|\theta_e^C)}, \tag{4}$$

where the probability of a category according to an entity's model ($P(c|\theta_e^C)$) and the probability of a category according to the query model ($P(c|\theta_q^C)$) remain to be defined.

2.2 Modeling Entities

Term-based representation To estimate $P(t|\theta_e^T)$ we smooth the empirical entity model with the background collection to prevent zero probabilities. We employ Bayesian

smoothing using Dirichlet priors which has been shown to achieve superior performance on a variety of tasks and collections [7, 5] and set:

$$P(t|\theta_e^T) = \frac{n(t,e) + \mu^T \cdot P(t)}{\sum_t n(t,e) + \mu^T}, \tag{5}$$

where $n(t,e)$ denotes the number of times t occurs in the document, $\sum_t n(t,e)$ is the total number of term occurrences, i.e., the document length, and $P(t)$ is the background model (the relative frequency of t in the collection). Since entities correspond to Wikipedia articles, this representation of an entity is identical to constructing a smoothed document model for each Wikipedia page, in a standard language modeling approach [6, 4]. Alternatively, the entity model can be expanded with terms from related entities, i.e., entities sharing the categories or entities linking to or from the Wikipedia page [3]. To remain focused, we do not explore this direction here.

Category-based representation Analogously to the term-based representation detailed above, we smooth the maximum-likelihood estimate with a background model. We employ Dirichlet smoothing again and use the parameter μ^C to avoid confusion with μ^T:

$$P(c|\theta_e^C) = \frac{n(c,e) + \mu^C \cdot P(c)}{\sum_c n(c,e) + \mu^C}. \tag{6}$$

In Eq. 6, $n(c,e)$ is 1 if entity e is assigned to category c, and 0 otherwise; $\sum_c n(c,e)$ is the total number of categories to which e is assigned; $P(c)$ is the background category model and is set using a maximum-likelihood estimate:

$$P(c) = \frac{\sum_e n(c,e)}{\sum_c \sum_e n(c,e)}, \tag{7}$$

where $\sum_c \sum_e n(c,e)$ is the number of category-entity assignments in the collection.

Entity priors. We use uniform entity priors, i.e., all pages in the collection are equally likely to be returned.

2.3 Modeling Queries

In this subsection we introduce methods for estimating and expanding query models. This boils down to estimating the probabilities $P(t|\theta_q^T)$ and $P(c|\theta_q^C)$ as discussed in §2.1.

Term-based representation. The term-based component of the baseline query model is defined as follows:

$$P(t|\theta_q^T) = P_{bl}(t|\theta_q^T) = \frac{n(t,Q)}{\sum_t n(t,Q)}, \tag{8}$$

where $n(t,Q)$ stands for the number of times term t occurs in query Q.

The general form we use for expansion is a mixture of the baseline (subscripted with *bl*) defined in Eq. 8 and an expansion (subscripted with *ex*):

$$P(t|\theta_q^T) = (1 - \lambda^T) \cdot P_{bl}(t|\theta_q^T) + \lambda^T \cdot P_{ex}(t|\theta_q^T). \tag{9}$$

Given a set of feedback entities FB, the expanded query model is constructed as follows:

$$P_{ex}(t|\theta_q^T) = \frac{P_{K_T}(t|FB)}{\sum_{t'} P_{K_T}(t'|FB)}, \tag{10}$$

where $P_{K_T}(t|FB)$ is estimated as follows. First, $P(t|FB)$ is computed according to Eq. 11. Then, the top K_T terms with the highest $P(t|FB)$ value are taken to form $P_{K_T}(t|FB)$, by redistributing the probability mass in proportion to their corresponding $P(t|FB)$ values:

$$P(t|FB) = \frac{1}{|FB|} \sum_{e \in FB} \frac{s(t,e)}{\sum_t s(t,e)} \tag{11}$$

and

$$s(t,e) = \log\left(\frac{n(t,e)}{P(t) \cdot \sum_t n(t,e)}\right), \tag{12}$$

where $\sum_t n(t,e)$ is the total number of terms, i.e., the length of the document corresponding to entity e. (This is the same as the *EXP* query model generation method using example documents from [1], with the simplification that all feedback documents are assumed to be equally important.)

The set of feedback entities, FB, is defined in two ways: for the entity ranking task, it is the top N relevant entities according to a ranking obtained using the initial (baseline) query. For the list completion task, the set of example entities provided with the query are used as the feedback set ($FB = E$).

Category-based representation. Our baseline model uses the keyword query (Q) to infer the category-component of the query model (θ_q^C), by considering the top N_c most relevant categories given the query; relevance of a category is estimated based on matching between the name of the category and the query, i.e., a standard language modeling approach on top of an index of category names:

$$P(c|\theta_q^C) = P_q(c|\theta_q^C) = \begin{cases} P(Q|\theta_c) / \sum_{c \in N_c} P(Q|\theta_c), & \text{if } c \in \text{top } N_c \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Note that this method does not use the category information provided with the query. To use target category information, we set $n(c,q)$ to 1 if c is a target category, and $\sum_c n(c,q)$ to the total number of target categories provided with the topic statement. Then, we put

$$P_c(c|\theta_q^C) = \frac{n(c,q)}{\sum_c n(c,q)}. \tag{14}$$

To combine the two methods (categories relevant to the query and categories provided as input), we put:

$$P(c|\theta_q^C) = \frac{1}{2} P_q(c|\theta_q^C) + \frac{1}{2} P_c(c|\theta_q^C). \tag{15}$$

(For the sake of simplicity, each model contributes half of the probability mass.)

Expansion of the category-based component is performed similarly to the term-based case; we use a linear combination of the baseline (either Eq. 13 or Eq. 15) and expanded components:

$$P(c|\theta_q^C) = (1 - \lambda^C) \cdot P_{bl}(c|\theta_q^C) + \lambda^C \cdot P_{ex}(c|\theta_q^C). \tag{16}$$

Given a set of feedback entities FB, the expanded query model is constructed as follows:

$$P_{ex}(c|\theta_q^C) = \frac{P_{K_C}(c|FB)}{\sum_{c'} P_{K_C}(c'|FB)}, \tag{17}$$

where $P_{K_C}(c|FB)$ is calculated similarly to the term-based case: first, $P(c|FB)$ is calculated according to Eq. 18 (where, as before, $n(c, e)$ is 1 if e belongs to c). Then, the top K_C categories with the highest $P(c|FB)$ value are selected, and their corresponding probabilities are renormalized, resulting in $P_{K_C}(c|FB)$.

$$P(c|FB) = \frac{1}{|FB|} \sum_{e \in FB} \frac{n(c, e)}{\sum_t n(c, e)}. \tag{18}$$

The set of feedback entities is defined as before (the top N entities obtained using blind relevance feedback for entity ranking, and the example entities E for list completion).

2.4 Heuristic Promotion

This year's topics are a selection of topics from prior editions of the track (from 2007 and 2008). We experiment with making use of the relevance assessments available from prior years; we view this information as click-through data. For so-called repeat queries such data can easily be collected in an operational system, thus assuming its availability is not unreasonable.

We incorporate this information into our ranking by mixing the original query likelihood probability ($P(q|e)$) with query likelihood based on click-through data ($P_c(q|e)$):

$$P'(q|e) = (1 - \alpha) \cdot P(q|e) + \alpha \cdot P_c(q|e). \tag{19}$$

For the sake of simplicity we set α in Eq. 19 to 0.5.

Let $c(q, e)$ denote the number of clicks entity e has received for query q; we set $c(q, e)$ to 1 if the entity was judged relevant for the query on the 2007 or 2008 topic sets, otherwise we set it to 0. We use a maximum likelihood estimate to obtain $P_c(q|e)$:

$$P_c(q|e) = \frac{c(q, e)}{\sum_{e'} c(q, e')}, \tag{20}$$

where the term $\sum_{e'} c(q, e')$ denotes the total number of clicks that query q has received (in our setting: the total number of entities judged as relevant for q).

Note that the underlying Wikipedia crawl has changed (from ∼659K to ∼2,666K documents), therefore this method is expected to increase early precision, but does not affect recall.

3 Runs

Parameter settings. Using the 2007 and 2008 editions of the Entity Ranking track as training material, we set the parameters of our models as follows.

- Importance of the term-based vs. the category-based component (Eq. 2): $\lambda = 0.7$
- Number of categories obtained given the query (Eq. 13): $N_c = 15$
- Number of feedback entities: $N = 3$
- Number of feedback terms (Eq. 17): $K_T = 35$
- Weight of feedback terms (Eq. 9): $\lambda^T = 0.7$
- Number of feedback categories (Eq. 17): $K_C = \infty$ (not limited)
- Weight of feedback categories (Eq. 16): $\lambda^C = 0.3$

Entity ranking. Table 1 summarizes the 4 runs we submitted for the entity ranking task. The baseline query models are estimated in the same manner for all runs: using Eq. 8 for the term-based component ($P(t|\theta_q^T)$) and Eq. 15 for the category-based component ($P(c|\theta_q^C)$). Runs that employ feedback estimate the term-based and category-based components of expanded query models using Eq. 10 and Eq. 17, respectively. The last two runs apply blind feedback only on a selection of topics; on those that were helped by this technique in prior editions of the task. Note that expansion always takes place in both the term-based and category-based components.

Table 1. Entity ranking runs. Feedback values are: N=no feedback, B=blind feedback, TB=topic-dependent blind feedback (only for topics that were helped by blind feedback on the 2007/2008 topic set.)

RunID (UAmsISLA_ER_...)	Feedback	Promotion	xinfAP
TC_ERbaseline	N	N	0.1893
TC_ERfeedback	B	N	0.2093
TC_ERfeedbackS	TB	N	0.2094
TC_ERfeedbackSP	TB	Y	0.5046

Table 2. List completion runs. (*Topics that were helped by using example entities on the 2007/2008 topic set do not use input category information (i.e., use Eq. 13 for constructing $P_{bl}(c|\theta_q^C)$); the remainder of the topics use the input category information (i.e., $P_{bl}(c|\theta_q^C)$ is estimated using Eq. 15).)

RunID (UAmsISLA_LC_...)	Expansion		Promotion	xinfAP
	Term	Cat.		
TE_LCexpT	Y	N	N	0.3198
TE_LCexpC	N	Y	N	0.2051
TE_LCexpTC	Y	Y	N	0.4021
TE_LCexpTCP	Y	Y	Y	0.5204
TEC_LCexpTCS*	Y	Y	N	0.3509
TEC_LCexpTCSP*	Y	Y	Y	0.5034

List completion. Table 2 summarizes the 6 runs we submitted for the list completion task. The baseline query models are estimated as follows: using Eq. 8 for the term-based component and Eq. 13 for the category-based component. The first four runs use only example entities (E), while the last two runs, that employ selective blind feedback (TEC_LCexpTCS and TEC_LCexpTCSP), also use input category information (C) for constructing the category-based component of the baseline query model (Eq. 15). We make use of example entities (E) in the feedback phase, by expanding the term-based and/or category-based component of query models with information extracted from examples (using Eq. 10 and Eq. 17 for term-based and category-based expansion, respectively).

4 Results and Discussion

Tables 1 and 2 present the results for the entity ranking (ER) and list completion (LC) tasks. Our main findings are as follows. First, feedback improves performance on both tasks; not surprisingly, explicit feedback (using example entities) is more beneficial than blind feedback. As to term-based vs. category-based feedback, the former is more effective on the LC task (TE_LCexpT vs. TE_LCexpC). This finding is especially interesting, as in previous work category-based feedback was found to be more advantageous [2]. This is probably due to the different term importance weighting scheme that we employed in the current work, which seems to be capable of sampling more meaningful terms from example entities. Combining term-based and category-based feedback improves over each method individually (TE_LCexpTC vs. TE_LCexpC and TE_LCexpTC vs. TE_LCexpT); this is in line with findings of [2], although the degree of improvement is more substantial here.

We made use of information from prior editions of the track in various ways. Applying blind feedback only to a subset of topics, that are likely to be helped, does not lead to meaningful score differences (TC_ERfeedback vs. TC_ERfeedbackS). Using explicit category information for a selected subset of topics hurt performance on the LC task (TE_LCexpTC vs. TEC_LCexpTCS); this is rather unexpected and suggests that the category assignments underlying Wikipedia has changed in the new dump. Finally, we promoted entities which were previously known to be relevant given the query. This heuristic proved to be very effective for both tasks (see TC_ERfeedbackS vs. TC_ERfeedbackSP, TE_LCexpTC vs. TE_LCexpTCP, and TEC_LCexpTCS vs. TEC_LCexpTCSP).

5 Conclusions

We have described our participation in the INEX 2009 Entity Ranking track. Building on earlier work [2], we employed a probabilistic modeling framework for entity search, in which term-based and category-based representations of queries and entities are effectively integrated.

We submitted 4 runs for the entity ranking and 6 runs for the list completion tasks. Our main focus was on evaluating the effectiveness of our recently proposed entity retrieval framework on the new Wikipedia collection. We demonstrated that our approach

is robust, and that it delivers very competitive performance on this year's platform too. We experimented with various ways of making use of information from prior editions of the task, and found that using known relevant pages as click-through data has a very positive effect on retrieval performance.

One change we implemented to the original approach concerned the way in which the importance of feedback terms, sampled from example entities, is estimated. Using more discriminative terms proved advantageous, which suggests that there is more to be gained by developing alternative methods for estimating the importance of terms and categories to be sampled from example entities.

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, and by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

[1] Balog, K., Weerkamp, W., de Rijke, M.: A few examples go a long way: constructing query models from elaborate query formulations. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 371–378. ACM Press, New York (2008)

[2] Balog, K., Bron, M., de Rijke, M.: Category-based query modeling for entity search. In: 32nd European Conference on Information Retrieval (ECIR 2010), March 2010, pp. 319–331. Springer, Heidelberg (2010)

[3] Fissaha Adafre, S., de Rijke, M., Tjong Kim Sang, E.: Entity retrieval. In: Recent Advances in Natural Language Processing (RANLP 2007), September 2007, Borovets, Bulgaria (2007)

[4] Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 111–119. ACM, New York (2001)

[5] Losada, D., Azzopardi, L.: An analysis on document length retrieval trends in language modeling smoothing. Information Retrieval 11(2), 109–138 (2008)

[6] Song, F., Croft, W.B.: A general language model for information retrieval. In: CIKM '99: Proceedings of the eighth international conference on Information and knowledge management, pp. 316–321. ACM, New York (1999)

[7] Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems 22(2), 179–214 (2004)