Completing Lists of Entities

Sisa Fissaha Adafre*, Maarten de Rijke** &
Erik Tjong Kim Sang***

*Dublin City University, **University of Amsterdam
***University of Groningen

Abstract

We consider the list completion task, an entity retrieval task where, in return to a topic statement and a number of example entities, systems have to return further examples. For this task, we propose and evaluate several algorithms. One of the core challenges is to overcome the very limited amount of information that serves as input — to address this challenge we explore different representations of list descriptions. For evaluation purposes we make use of the lists and categories available in Wikipedia. Experimental results show that cluster-based contexts improve retrieval results.

1 Introduction

The information retrieval community and commercial systems and are both displaying an increasing interest in not just returning documents in response to a user's query but "objects," "entities" or their properties. E.g., various web search engines recognize specific types of entity (such as books, CDs, restaurants), and list these separately from the standard document-oriented hit list. Enterprise search provides another example (Craswell et al. 2001), as has also been recognized within the TREC Enterprise track. In its 2005 and 2006 editions, the track featured an expert finding task (Craswell et al. 2006) where systems return a list of entities (people's names) who are knowledgeable about a certain topic (e.g., "web standards").

This emerging area of *entity retrieval* differs from traditional document retrieval in a number of ways. Entities are not represented directly (as retrievable units such as documents), and we need to identify them "indirectly" through occurrences in documents. Entity retrieval systems may initially retrieve documents (pertaining to a given topic or entity) but they must then extract and process these documents in order to return a ranked list of entities (Petkova & Croft 2006). In order to understand the issues at hand, we consider one particular entity retrieval task (building on a proposal launched in the run-up to INEX 2006 (INEX 2006) and implemented at INEX 2007): *list completion.*

The *list completion* task is defined as follows. Given a topic text and a number of examples, the system has to produce further examples. I.e., given a topic description, a set of entities S and a number of example entities $e_1, \ldots, e_n$ in S that fit the description, return "more examples like $e_1, \ldots, e_n$" from S that fit the description. E.g., given the short description *tennis players* and two example entities such as *Kim Clijsters* and *Martina Hingis*, entities such as tennis tournaments or coaches are not relevant. Instead, the expected set should include only individuals who are or have been professional tennis players.

The main research questions we address concern the ways in which we represent entities and in which we match topics and entities. As we will see, providing a sufficiently rich description of both topics and entities to be able to rank entities in an effective manner, is one of the main challenges. We address this challenge by using several contextual models.

For evaluation purposes we make use of Wikipedia, the online encyclopedia. The decision for using Wikipedia for this task is based on practical and theoretical considerations. Wikipedia contains a large set of lists that can be used for generating the necessary test data, and also assessing the outputs of our methods. Also, with its rich structure Wikipedia offers an interesting experimental setting where we can experiment with different features, both content-based and structural. Finally, by using Wikipedia's lists, we can avoid the information extraction task of *identifying entities* in documents and focus on the retrieval task itself, instead. Below, we will only consider entities available in Wikipedia, and we will identify each entity with its Wikipedia article.[1]

The remainder of the paper is organized as follows. First, we provide background material and related work on working with Wikipedia, list questions, and contextual models. After that we turn to the list completion task, proposing and evaluating a number of algorithms. We end with a discussion and conclusion.

2 Background

Mining/Retrieval against Wikipedia. Wikipedia has attracted interest from researchers in disciplines ranging from collaborative content development to language technology, addressing aspects such as information quality, users motivation, collaboration pattern, network structures, e.g., (Zlatić et al. 2006). Several publications describe the use of Wikipedia as a resource for question answering and other types of IR systems; see e.g., (Ahn et al. 2006, Fissaha Adafre & de Rijke 2007a, Jijkoun & de Rijke 2007). Wikipedia has been used for computing word semantic relatedness, named-entity dis-

[1] We used the XML version of the English Wikipedia corpus made available by (Denoyer and Gallinari 2007). It contains 659,388 articles, and has annotations for structural elements such as article title, sections, paragraphs, sentences, and hyperlinks.

ambiguation, text classification, and in various retrieval and knowledge representation tasks, e.g., (Gabrilovich & Markovitch 2006).

Entity retrieval. List queries are a common types of web queries (Rose & Levinson 2004). The TREC Question Answering track has recognized the importance of list questions (Voorhees 2005); there, systems have to return two or more instances of the class of entities that match the description in the list question. List questions are often treated as (repeated) factoids, but special strategies are called for as answers may need to be collected from multiple documents (Chu-Carroll et al. 2004).

Recognizing the importance of list queries, Google Sets allows users to enter some instances of a concept and retrieve others that closely match the examples provided (Google 2006). Ghahramani & Heller (2005) developed an algorithm for completing a list based on examples using machine learning techniques. A proposed INEX entity retrieval track, with several tasks, was run in 2007 and will run again in 2008 (INEX 2006).

Our entity retrieval tasks are related to ontological relation extraction (Hearst 1992), where a combination of large corpora with simple manually created patterns are often used. Wikipedia, as a corpus, is relatively small, with much of the information being presented in a concise and non-redundant manner. Therefore, pattern-based methods may have limited coverage for the entity retrieval tasks that we consider.

Document expansion and contextual IR. Enriching the document representation forms an integral part of the approach we propose. Though, in the past, application of document expansion techniques, particularly document clustering, has shown mixed results in document retrieval settings, recent studies within the language modelling framework provide new supporting evidence of the advantages of using document clusters (Liu & Croft 2004). Due to the nature of the tasks defined in this paper, the cluster hypothesis which states that "closely associated documents tend to be relevant to the same request" (Jardine & van Rijsbergen 1971) provides for an intuitive starting point in designing our methods. Specifically, for each entity (or article) a precomputed cluster will be used to supply it with contextual information, much in the spirit of the work done by (Azzopardi 2006).

3 List completion

The main challenge of the list completion task is that the topic statement, example entity descriptions, and, more generally, entity descriptions in Wikipedia, tend to be very short. Therefore, a straightforward retrieval baseline may suffer from poor recall. Hence, in our modeling we will address several ways of representing the topic statement and example entities.

We model the list completion task as follows: *what is the probability of a candidate e belonging to the list defined by the topic statement t and example entities $e_1, \ldots, e_n$?* We determine $p(e|t, e_1, \ldots, e_n)$ and rank entities according to this probability. To estimate $p(e|t, e_1, \ldots, e_n)$, we proceed in two steps: (1) select candidate entities, and (2) rank candidate entities. More formally,

$$p(e|t, e_1, \ldots, e_n) \propto \chi_C \cdot \mathit{rank}(e; t, e_1, \ldots, e_n),$$

where χ_C is a characteristic function for a set of selected candidate entities C and $\mathit{rank}(\cdot)$ is a ranking function. Below, we consider alternative definitions of the function χ_C and we describe two ranking functions. First, though, we define so-called entity neighborhoods that will be used in the candidate selection phase: to each individual entity they associate additional entities based on its context, both in terms of link structure and contents.

3.1 *Entity neighborhoods*

In the context of a hypertext document, identification of a cluster typically involves searching for graph structures, where co-citations and bibliographic couplings provide importance features. Fissaha Adafre & de Rijke (2005) describe a Wikipedia specific clustering method called *LTRank*. Their clustering method primarily uses the co-citation counts. We provide a slight extension that exploits the link structure (both incoming and outgoing links), article structure, and content. In Wikipedia, the leading few paragraphs contain essential information about the entity being described in the articles serving as summary of the content of the article; we use the first five sentences of the Wikipedia article as a representation of the content of the article. Our extension of the LTRank method for finding the neighborhood $\mathit{neighborhood}(e)$ of an entity e is summarized in Figure 1. With this definition we can turn to the first phase in our approach: *candidate entity selection*.

3.2 *Candidate entity selection*

To perform the candidate entity selection step, we use a two part representation of entities (Wikipedia articles). Each entity e is represented using (1) the textual content of the corresponding article a_e, and (2) the list of all entities in the set of $\mathit{neighborhood}(e)$ defined above. We propose four candidate entity selection methods, B-1, ..., B-4, that exploit this representation in different ways.

- Given a Wikipedia article a_e of an entity e, collect the titles of pages with links to or from a_e, as well as the words in the first five sentences of a_e. Let $long(a_e)$ be the resulting bag of terms; this is the *long* representation of a_e.
- Given a Wikipedia article a_e, rank all articles w.r.t. their content similarity to $long(a_e)$; we use a simple vector space model for the ranking. This produces a ranked list $L_{a_e} = a_{e_1}, \ldots, a_{e_n}, \ldots$.
- Given a Wikipedia article a_e, consider the titles $t_1, \ldots, t_k$ of the top k articles in the list L_{a_e}. Represent a_e as the bag of terms $short(a_e) = \{t_1, \ldots, t_k\}$; we call this the *short* representation of a_e.
- For each Wikipedia article a_e, rank the short representations of other Wikipedia articles w.r.t. their content similarity to $short(a_e)$; again, we use a simple vector space model for the ranking. This produces a ranked list L'_{a_e}. The $neighborhood(e)$ is defined to be the set of top l articles in L'_{a_e} whose similarity score is above some threshold α.

Fig. 1: *An extension of LTRank (Fissaha Adafre & de Rijke 2005). Our extension is in the first step, where we add outgoing links and the first 5 sentences of* a_e. *In this paper, we took* $k = 10$, $l = 100$ *and* $\alpha = 0.3$

B-1. Baseline: Retrieval. Here we rank entities by the similarity of their content part to a query consisting of the topic statement t and the titles $t_{e_1}, \ldots, t_{e_n}$ of the example entities. We used a simple vector space retrieval model for computing the similarity. The top n retrieved documents constitute the baseline candidate set C_1.

B-2. Neighborhood search. Our second candidate selection method matches the titles of the example entities against the neighborhoods of Wikipedia articles.

$$C_2 = \{e | \bigvee_i (e_i \in neighborhood(e))\}$$

B-3. Neighborhood and Topic statement search. Here we take the union of the entities retrieved using the topic statement, and method B-2 described above. First, we rank entities by the similarity of their content part to a query which corresponds to the topic statement t. Here again, we used a simple vector space similarity measure to compute the similarity. We take the top k entities ($k = 200$ in this paper) which constitute the first set, $C_{3.1}$. We then take all entities that contain at least one example entity in their neighborhood as with B-2, i.e.,

$$C_{3.2} = \{e | \bigvee_i (e_i \in neighborhood(e))\}.$$

The final candidate set is simply the union of these two sets, i.e., $C_3 = C_{3.1} \cup C_{3.2}$.

B-4. Neighborhood and Definition search. This method is similar to the method B-3. But instead of taking the topic statement t as a query for ranking entities (in the set $C_{3.1}$ above), we take the definitions of the example entities $e_1, \ldots, e_n$, where the first sentence of the Wikipedia article a_e of an entity e to be its definition; stopwords are removed.

3.3 *Candidate entity ranking*

We compare two methods that make use of the content of articles for ranking the entities generated by the previous step. Particularly, we apply the following two methods: Bayesian inference (Ghahramani & Heller 2005) and relevance-based language models (Lavrenko & Croft 2003). Both methods provide a mechanism for building a model of the concept represented by the example set. These two algorithms are developed for a task which closely resembles our task definitions, i.e., given a limited set of examples, find other instances of the concept represented by the examples. In the next paragraphs, we briefly discuss these methods.

C-1. Bayesian Inference. Ghahramani & Heller (2005) addressed the entity ranking task in the framework of Bayesian Inference. Given n example entities, $e_1, \ldots, e_n$, and candidate entity e, the ranking algorithm is given by

$$score(e) = \frac{P(e, e_1, \ldots, e_n)}{P(e)\, P(e_1, \ldots, e_n)}. \tag{1}$$

To compute Eq. 1, a parameterized density function is posited. We list all terms $t_{e_{1,1}}, \ldots, t_{e_{1,k_1}}, \ldots, t_{e_{n,k_n}}$ occurring in the example entities. Then, each candidate entity e is represented as a binary vector where vector element $e_{i,j}$ corresponds to the j-th term from article a_{e_i} of the i-th example instance and assumes 1 if $t_{e_{i,j}}$ appears in the article for the entity e and 0 otherwise. It is assumed that the terms $e_{i,j}$ are independent and have a Bernoulli distribution θ_j with parameters α_j and β_j; see (Ghahramani & Heller 2005). In sum, Eq. 1 is rewritten to:

$$score(e) = c + \textstyle\sum_{j=1}^{N} q_j e_{\cdot,j},$$

where the summation ranges over the binary vector representation of e, and

$$\begin{aligned} c \;&=\; \textstyle\sum_j \big(\log(\alpha_j + \beta_j) - \log(\alpha_j + \beta_j + n) + \\ &\quad \log(\beta_j + n - \textstyle\sum_{i=1}^{n} e_{i,j}) - \log(\beta_j)\ \big)\,, \end{aligned}$$

while

$$\begin{aligned} q_j \;&=\; \log(\alpha_j + \textstyle\sum_{i=1}^{n} e_{i,j}) - \log(\alpha_j) + \\ &\quad \log(\beta_j) - \log(\beta_j + n - \textstyle\sum_{i=1}^{n} e_{i,j}) \end{aligned}$$

For given values of α_j and β_j, the quantity q_j assigns more weight to terms that occur in most of the example entities. Therefore, a candidate instance e_i will be ranked high if it contains many terms from the example instances and the $e_{i,j}$ receive high weights from the q_js.

C-2. Relevance Models. Lavrenko & Croft (2003) proposed so-called relevance-based language models for information retrieval. Given n example entities, $e_1, \ldots, e_n$, and the candidate e from the candidate set C, the ranking function is given by the KL-divergence between two relevance models:

$$score(e) = KL(P_{e_1,\ldots,e_n} || P_e),$$

where $P_{e_1,\ldots,e_n}$ is the relevance model of the example entities, and P_e is the language model induced from the Wikipedia article for entity e. The relevance models are given by

$$\begin{aligned}
P(w|e_1, \ldots, e_n) &= \textstyle\sum_{e \in W} P(w|e) \cdot P(e|e_1, \ldots, e_n) \\
P(e|e_1, \ldots, e_n) &= \begin{cases} 1/n & \text{if } e \in \{e_1, \ldots, e_n\} \\ 0 & \text{otherwise} \end{cases} \\
P(w|e) &= \frac{\#(w,e)}{|e|},
\end{aligned}$$

where W is the collection (Wikipedia), and w represents the terms in the Wikipedia article for entity e. The KL divergence will be small for entities that more closely resemble the example entities in terms of their descriptions.

Summary. Both of the ranking methods outlined above return a ranked list of candidate entities. We normalize the scores using

$$\text{score}_{\text{norm}} = \frac{\text{score}_{\text{MAX}} - \text{score}}{\text{score}_{\text{MAX}} - \text{score}_{\text{MIN}}},$$

and take those candidate entities for which the normalized score lie above an empirically determined threshold ($\text{score}_{\text{norm}} > 0.5$). The resulting set constitutes our output.

3.4 *Experimental set-up*

The performance of our approach to the list completion task depends on the performance of the two subcomponents: candidate selection and candidate ranking. We conduct two sets of experiments, one to determine the effectiveness of the candidate selection methods, and a second to determine the effectiveness of the overall approach. We are especially interested in the contribution of using the neighborhoods of entities.

The Wikipedia lists serve as our gold standard. We selected a random sample of 30 lists (the topics) from Wikipedia. We chose relatively homogeneous and complete lists, and excluded those that represent a mixture of several concepts. We take 10 example sets for each topic. Each example set consists of a random sample of entities from the Wikipedia list for the topic. We run our system using each of these 10 example sets as a separate input. The final score for each topic is then the average score over the ten separate runs. In the experiments in this section, we assume that each example set contains two example instances. This choice is mainly motivated by our assumption that users are unlikely to supply many examples.

The results are assessed based on the following scores: *P@20* (number of correct entities that are among the top 20 in the ranked list), *precision* (P; number of correct entities that are in the ranked list, divided by the size of the ranked list), *recall* (R; number of correct entities that are in the ranked list, divided by the number of entities in the Wikipedia list) and *F-score* (F; harmonic mean of the recall and precision values).

In order to test if the differences among the methods measured in terms of F-scores is statistically significant, we applied the two-tailed Wilcoxon matched pair signed-ranks test (for $\alpha = 0.05$ and $\alpha = 0.005$).

3.5 *Results*

First, we assess the methods we used for candidate selection. Following this, we present the evaluation results of the overall system.

Candidate selection. Table 1 shows results of the evaluation of the candidate selection module. The figures are averages over all topics and all sets of example entities. The values are relatively low. Retrieving additional candidates using terms derived either from the definition of the entities or topic statement improves recall to some extent. The recall values for method B-3 are the best. This suggests that the terms in the topic are more accurate than the terms automatically derived from the definitions.

The neighborhood-based methods achieve better recall values while returning fewer number of candidates (cf. the last column of Table 1).

Overall results. Table 2 shows the scores resulting from applying the two ranking methods C-1 and C-2 on the output of different candidate selection methods. The first column of Table 2 shows the different candidate selection methods; the second column shows the ranking methods.

Selection method	P	R	Result set size
B-1 (Top $k = 500$)	0.042	0.235	500
B-2	0.142	0.236	206
B-3	0.089	0.311	386
B-4	0.093	0.280	367

Table 1: *Performance on the candidate selection subtask*

Candidate selection	ranking	P	R	F	P@20
B-1	C-1	0.100	0.068	0.058	0.128
	C-2	0.203	0.046	0.060	0.144
B-2	C-1	0.172	0.163	0.136	0.205
	C-2	**0.227**	0.142	0.137	0.231
B-3	C-1	0.121	**0.236**	0.136	0.196
	C-2	0.188	0.210	0.151	**0.249**
B-4	C-1	0.140	0.202	0.142	0.201
	C-2	0.204	0.209	**0.158**	0.248

Table 2: *Performance on the entire list completion task. Best scores per metric in boldface*

The neighborhood-based combinations outperform the baselines at the $\alpha = 0.005$ significance level (when considering F-scores). The combination of C-2 (*Relevance model*) with B-4 (*Neighborhood plus Definition Terms*) input outperforms both the B-2 + C-1 and B-2 + C-2 combinations at the $\alpha = 0.05$ significance level. Generally, the C-2 ranking method has a slight edge over the C-1 method on most inputs. Furthermore, retrieving additional candidates using either the topic statement or the definition terms improves results, especially when used in combination with the C-2 ranking method.

3.6 *Error analysis*

A closer look at the results for individual topics reveals a broad range of recall values. The recall values for the topics *North European Jews*, *Chinese Americans*, *French people*, and *Miami University alumni* are very low. On the other hand, the topics *Indian Test cricketers*, *Revision control software*, *Places in Norfolk*, and *Cities in Kentucky* receive high recall scores. For the neighborhood-based methods, there is some correlation between the composition of the neighborhoods corresponding to the example entities and the results obtained. E.g., the neighborhoods corresponding to the example entities for the topic *Indian Test cricketers* contain Indian cricket players. On the other hand, the neighborhoods corresponding to the example entities for the topic *Chinese Americans* contain individuals from the USA, most of whom are not Chinese Americans, and have very little in common except for the features identified by the topic titles, which are too specific.

4 Discussion

List completion aims at identifying entities that share certain characteristics. In this respect, it resembles tasks commonly addressed in Information Extraction (IE), such as *named entity recognition* and *relation extraction*. However, there are important distinctions between traditional IE and the list completion task. First, in typical IE scenarios, the entities are embedded in a text, and the aim is to extract or recognise occurrences of these entities in the text. Systems commonly use surrounding contextual information, and redundancy information to do this. The inputs to these systems are documents that may contain one or more occurrences of the target entities. In contrast, in list completion, the entities are represented by documents which provide descriptive information about them — typically, there is a one-to-one relation between the entities and the documents. In our setting, then, we abstract away from the recognition phase so that we are able to zoom in on the retrieval task only — unlike, e.g., the expert finding scenarios currently being explored at TREC, that do require participating systems to create effective combinations of extraction and retrieval (Balog & de Rijke 2006).

At an abstract level, the list completion task is similar to the so-called *entity ranking* task where a system has to return entities that satisfy a topic described in natural language text (Fissaha Adafre & de Rijke 2007b). But a closer look reveals important differences which necessitates task-specific approaches. One aspect concerns the size of the input; for the list completion task, the inputs are example entities with/without topic statements, and the candidates are all Wikipedia entries. On the other hand, the inputs for the entity ranking task consist of topic statements only, and the candidates are entities in a particular Wikipedia list, such as, e.g., the List of Countries, which is obviously much smaller and more homogeneous than the entire Wikipedia collection.

Our results show that traditional information retrieval methods significantly underperform for selecting initial candidates from all of Wikipedia. This affects the overall score of the method as subsequent processing makes use of the output of this step. On the other hand, preclustering of Wikipedia articles led to much better performance. The re-ranking methods showed comparable performance results, with the relevance feedback method having a slight edge over the Bayesian method.

5 Conclusion

We described, and proposed solutions for, an entity retrieval task, viz. *list completion*. We conducted a sets of experiments in order to assess the proposed methods, which focused on enriching the two key elements of the retrieval tasks, i.e., *Topic statements* and *Example entities*.

The methods that used the titles of the example entities and the topic statements or definition terms performed better. All methods that used a context set consisting of related articles significantly outperformed a document-based retrieval baseline that does not use the related articles field.

Our results are limited in several ways. E.g., entities are represented primarily by the combination of the content of their Wikipedia articles (as a bag of words) and a precomputed set of related articles. We need to explore other—rich—representations of the content, e.g., phrases or anchor text, and also other concepts of relatedness, e.g., the Wikipedia categories.

Acknowledgments. This research was supported by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104, the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (`http://www.stevin-tst.org`) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 017.001.190, 640.001.501, 640.002.-501, 612.066.512, 612.061.814, 612.061.815.

REFERENCES

Ahn, K., J. Bos, J.R. Curran, D. Kor, M. Nissim & B. Webber. 2005. "Question Answering with QED at TREC-2005". *Proceedings of TREC 2005*. NIST Special Publication SP 500-266, Washington, USA.

Azzopardi, L. 2006. "Incorporating Context within the Language Modeling Approach for ad hoc Information Retrieval". *SIGIR Forum* 40:1.70-70.

Balog, K. & M. de Rijke. 2006. "Finding Experts and Their Details in E-mail Corpora". *15th International World Wide Web Conference* (*WWW2006*), 1035–1036. Edinburgh, Scotland, UK.

Chu-Carroll, J., K. Czuba, J. Prager, A. Ittycheriah & S. Blair-Goldensohn. 2004. "IBM's PIQUANT II in TREC 2004". *Proceedings TREC 2004*. NIST Special Publication 500-261, Washington, USA.

Craswell, N., D. Hawking, A. M. Vercoustre & P. Wilkins. 2001. "P@noptic Expert: Searching for Experts not just for Documents". *Ausweb*. Coffs Harbour, Australia.

Craswell, N., A. de Vries & I. Soboroff. 2006. "Overview of the TREC 2005 Enterprise Track". *Proceedings of TREC 2005*. NIST Special Publication

SP 500-266, Washington, USA.

de Vries, A. & N. Craswell. 2006. "XML Entity Ranking Track" — `http://inex.is.informatik.uni-duisburg.de/2006/xmlSearch.html` [Source checked in May 2008]

Denoyer, L. & P. Gallinari. 2007. "The Wikipedia XML Corpus". *INEX 2006*, 10–16. Dagstuhl, Germany.

Fissaha Adafre, S. & M. de Rijke. 2005. "Discovering missing links in Wikipedia". *Proceedings of LinkKDD-2005 Workshop*. Chicago, IL., USA.

Fissaha Adafre, S. & M. de Rijke. 2007a. "Estimating Importance Features for Fact Mining (with a Case Study in Biography Mining)". *RIAO 2007*. Pittsburgh, USA.

Fissaha Adafre, S. & M. de Rijke. 2007b. "Entity Retrieval". *International Conference on Recent Advances in Natural Language Processing* (*RANLP-2007*) ed. by G.Angelova, K.Bontcheva, R.Mitkov, N.Nicolov & N.Nikolov. Borovets, Bulgaria.

Gabrilovich, E. & S. Markovitch. 2006. "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge". *AAAI '06*. Boston, Massachusetts, USA.

Ghahramani, Z. & K. A. Heller. 2005. "Bayesian Sets". *NIPS 2005*, 435–442. Cambridge, MA., USA.

Google. 2006. GoogleSets — `http://labs.google.com/sets` [Source checked in Oct. 2004]

Hearst, M. A. 1992. "Automatic Acquisition of Hyponyms from Large Text Corpora". *Proceedings of COLING 1992*, 539-545. Nantes, France.

Jardine, N. & C. J. van Rijsbergen. 1971. "The Use of Hierarchic Clustering in Information Retrieval". *Information Storage Retrieval* 7:5.217-240.

Jijkoun, V. & M. de Rijke. 2007. "WiQA: Evaluating Multi-lingual Focused Access to Wikipedia". *EVIA 2007*, Tokyo, Japan.

Lavrenko, V. & W. B. Croft. 2003. "Relevance Models in Information Retrieval". *Language Modeling for Information Retrieval*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Liu, X. & W. B. Croft. 2004. "Cluster-based Retrieval Using Language Models". *SIGIR '04*, 186-193. Sheffield, UK.

Petkova, D. & W. B. Croft. 2006. "Hierarchical Language Models for Expert Finding in Enterprise Corpora". *Proceedings ICTAI 2006*, 599-608. Washington, USA.

Rose, D. E. & D. Levinson. 2004. "Understanding User Goals in Web Search". *WWW '04*, 13–19. New York, USA.

Voorhees, E. 2005. "Overview of the TREC 2004 Question Answering Track". *Proceedings of TREC 2004*. NIST Special Publication: SP 500-261, Washington, USA.

Zlatić, V., M. Božičević, H. Štefančić & M. Domazet. 2006. "Wikipedias: Collaborative Web-based Encyclopedias as Complex Networks". *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* 74:016115, 2006.