Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah

Theodora Tsikrika[1], Pavel Serdyukov[2], Henning Rode[2], Thijs Westerveld[3]*, Robin Aly[2], Djoerd Hiemstra[2], and Arjen P. de Vries[1]

[1] CWI, Amsterdam, The Netherlands
[2] University of Twente, Enschede, The Netherlands
[3] Teezir Search Solutions, Ede, The Netherlands

Abstract. CWI and University of Twente used PF/Tijah, a flexible XML retrieval system, to evaluate structured document retrieval, multimedia retrieval, and entity ranking tasks in the context of INEX 2007. For the retrieval of textual and multimedia elements in the Wikipedia data, we investigated various length priors and found that biasing towards longer elements than the ones retrieved by our language modelling approach can be useful. For retrieving images in isolation, we found that their associated text is a very good source of evidence in the Wikipedia collection. For the entity ranking task, we used random walks to model multi-step relevance propagation from the articles describing entities to all related entities and further, and obtained promising results.

1 Introduction

In INEX 2007, CWI and the University of Twente participated in the Ad Hoc, Multimedia, and Entity Ranking tracks. In all three tracks, we used PF/Tijah [5], a flexible system for retrieval from structured document collections, that integrates NEXI-based IR functionality and full XQuery support.

In the Ad Hoc track, we participated in all three subtasks for element retrieval, and mainly investigated the effect of various length priors within a language modelling framework. We also took part in both Multimedia tasks, where we examined the value of textual and context-based evidence without considering any of the available visual evidence. For Entity Ranking, we exploited the associations between entities; entities are ranked by constructing a query-dependent entity link graph and applying relevance propagation schemes modelled by random walks.

The remainder of this paper is organised as follows. Section 2 introduces PF/Tijah. Next, Sections 3, 4, and 5 respectively discuss our participation in each of the Ad Hoc, Multimedia, and Entity Ranking tracks. Section 6 concludes this paper by highlighting our main contributions.

* This work was carried out when the author was at CWI, Amsterdam, The Netherlands

2 The PF/Tijah System

PF/Tijah, a research project run by the University of Twente, aims at creating a flexible environment for setting up search systems. It achieves that by including out-of-the-box solutions for common retrieval tasks, such as index creation (that also supports stemming and stopword removal) and retrieval in response to structured queries (where the ranking can be generated according to any of several retrieval models). Moreover, it maintains its versatility by being open to adaptations and extensions.

PF/Tijah is part of the open source release of MonetDB/XQuery (available at `http://www.sourceforge.net/projects/monetdb/`), which is being developed in cooperation with CWI, Amsterdam and the University of München. PF/Tijah combines database and information retrieval technologies by integrating the PathFinder (PF) XQuery compiler [1] with the Tijah XML information retrieval system [11]. This provides PF/Tijah with a number of unique features that distinguish it from most other open source information retrieval systems:

- It supports retrieval of arbitrary parts of XML documents, without requiring a definition at indexing time of what constitutes a document (or document field). A query can simply ask for any XML tag-name as the unit of retrieval without the need to re-index the collection.
- It allows complex scoring and ranking of the retrieved results by directly supporting the NEXI query language.
- It embeds NEXI queries as functions in the XQuery language, leading to ad hoc result presentation by means of its query language.
- It supports text search combined with traditional database querying.

The above characteristics also make PF/Tijah particularly suited for environments like INEX, where search systems need to handle highly structured XML collections with heterogenous content. Information on PF/Tijah, including usage examples, can be found at: `http://dbappl.cs.utwente.nl/pftijah/`.

3 Ad Hoc Track

The granularity at which to return information to the user has always been an important aspect of the INEX benchmarks. The element and passage retrieval tasks aim to study ways of pointing users to the most specific relevant parts of documents. Various characteristics of the document parts or elements are of potential value in identifying the most relevant retrieval bits. Obviously the element content is a valuable indicator, but also more superficial features like the element type, the structural relation to other elements and the depth of the XML tree may play a role.

We studied the influence of a very basic feature: element size. Size priors have played an important role in information retrieval [14, 4, 8]. Kamps et al. [6] studied length normalization in the context of XML retrieval and INEX collections and found that the size distribution of relevant elements differed significantly from the general size distribution of elements. Emphasizing longer elements by introducing, linear, quadratic or even cubic length priors improved the retrieval results significantly on the IEEE collection.

For this paper, we experimented with biasing towards longer elements (similarly to Kamps et al. [6]), but in the setting of the Wikipedia collection. We use a language modelling framework where document priors are incorporated as a priori probabilities of relevance based on document characteristics that are independent of a query (element size in our case). The probability of a document D given a query Q can be factored as the probability of drawing the query from the document ($P(Q|D)$: the document's language model) and the prior probability of the document $P(D)$ (the prior probability of the query $P(Q)$ does not influence the ranking and can be ignored):

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \propto P(Q|D)P(D) \qquad (1)$$

where $P(Q|D)$ is estimated using a unigram language model smoothed by a Jelinek-Mercer parameter [4]. We also performed a retrospective study on the Wikipedia collection, where we analysed the size distributions of elements in the collection, in the relevant elements for the INEX 2006 Focused task, and in the elements retrieved by our baseline language model run.

3.1 Experiments with Length Priors

In our runs for INEX 2007, we experimented with different priors. We submitted runs with priors that are linear in the log of the element size (`star_logLP`) and runs with a normally distributed log size prior (`star_lognormal`). Each of the prior runs is submitted for the Focused task and in addition filtered for the Relevant in Context task (runIDs with `_Ric` affix); for relevant in context we grouped the results in a top 1500 baseline run by article and ordered the articles based on their top scoring element. In addition we submitted an article only baseline run, i.e. a run in which we only return full articles. This article run was submitted to both the Focused (`article`) and Best in Context tasks (`article_BiC`) . Tables 1, 2, and 3 show the results for these official submissions.

Table 1. Results for the CWI/UTwente submissions to the Ad Hoc Focused task. The table shows the rank of the run among official submissions, the run identifier and the interpolated precision at 0.01 recall.

rank	runID	iP[0.01]
56	star_logLP	0.3890
59	article	0.3701
78	star_lognormal	0.0381

Table 2. Results for the CWI/UTwente submissions to the Ad Hoc Relevant in Context task. The table shows the rank of the run among official submissions, the run identifier and Mean Average generalized precision.

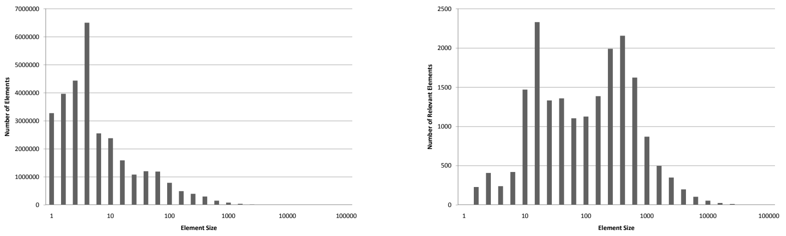
rank	runID	MAgP
15	star_logLP_RinC	0.1233
64	star_lognormal_RinC	0.0075

Table 3. Results for the CWI/UTwente submissions to the Ad Hoc Best in Context task. The table shows the rank of the run among official submissions, the run identifier and the Mean Average generalized precision.

rank	runID	MAgP
31	articleBic	0.1339

3.2 Analysis of Element Size

The disappointing results with the two size priors warrant a study of the distribution of element size in relevant and non-relevant elements. We studied INEX 2006 data to gain some insight. Figure 1 shows the distribution of element sizes in the Wikipedia collection as a whole and in the relevant elements. While the collection contains many small elements, these are rarely relevant. If we would not pay attention to element length and use a retrieval model that does not have a bias for elements of any size we would retrieve too many small elements. Simply giving a bias towards longer elements could improve retrieval results.



(a) XML element sizes

(b) Relevant element sizes

Fig. 1. Size distribution of collection elements and elements relevant to 2006 topics.

As previously mentioned, one way of compensating for this emphasis on small elements that nicely fits in the language modeling framework that we use is to incorporate document priors. The probability of relevance given a certain size can be estimated by comparing the distributions of relevant elements to those of the collection: $P_{size}(D) = P(relevant|size(D))$. This leads to the prior visualized in

Figure 2. A quadratic prior as found by Kamps et al. [6] for the IEEE collection seems appropriate.

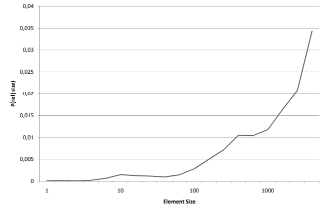
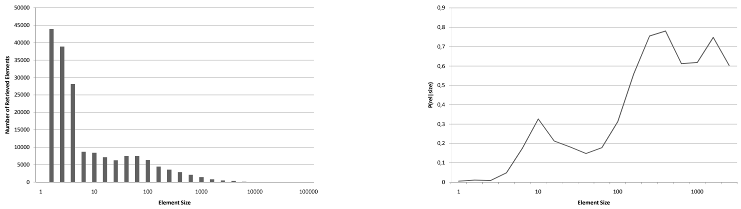


Fig. 2. Size prior estimated from INEX 2006 statistics for relevant and collection elements

However, in reality, a retrieval model does not retrieve elements of all sizes uniformly. For example, the language model we use interpolates document and collection probabilities in a standard manner and computes the document probability based on the relative frequency of query terms in documents [4]. This has the effect that short elements containing query terms get a high score. Figure 3a shows the distribution of elements that we retrieve using this language modeling approach if we do not compensate for document length. Clearly, we retrieve a lot of small elements.



(a) Size distribution of elements retrieved in Language Modeling framework

(b) Size prior estimated from the fraction of the number of relevant and retrieved elements

Fig. 3. Size distribution of retrieved elements and prior based on comparing this distribution with size distribution of relevant elements (Figure 1b)

To see which elements we should emphasize given the use of our language model, we also compute a prior based on comparing relevant to retrieved elements: $P_{size}(D) = P(relevant|size(D), retrieved(D))$. Figure 3b visualises these priors. Judging from this figure, it seems the prior should have a big peak around

1000 terms and a smaller peak around 10 terms. A mixture model seems an appropriate prior. Further experiments are needed to analyse the impact of such a prior on retrieval effectiveness.

4 Multimedia Track

CWI/Utwente participated in both MMfragments and MMimages tasks of the Multimedia track. Our overall aim is to investigate the value of textual and contextual evidence given information needs (and queries) with clear multimedia character. As a result, we only submitted text-based runs without taking into account any of the provided visual evidence. Below, we discuss our approaches and experimental results for both tasks.

4.1 MMfragments task

For MMfragments, the objective is to find relevant XML fragments (i.e., elements or passages) in the (Ad Hoc) Wikipedia XML collection given a multimedia information need. MMfragments is actually very similar to the Ad Hoc retrieval task, with the difference being that MMfragments has a multimedia character and, therefore, requires the retrieved fragments to contain at least one relevant image, together with relevant text. Furthemore, additional visual evidence, such as concepts and image similarity examples, can be provided as part of a topic. Given these similarities, MMfragments was run in conjunction with the Ad Hoc track, with MMfragments topics forming a subset of the Ad Hoc ones. In addition, MMfragments contains the same three substasks as the Ad Hoc task. This gives us the opportunity to compare the effectiveness of MMfragments runs (i.e., runs with a clear multimedia character) against Ad Hoc runs on the same topic subset.

We only participated in the Focused MMfragments task. Given the similarities with the Ad Hoc task, we decided to (i) use only the title field of the topics, (ii) apply the same three element runs as the ones submitted for the Focused Ad Hoc task (i.e., `article`, `star_logLP` and `star_lognormal`), and (iii) realise the multimedia character by filtering our results, so that we only return fragments that contain at least one image. Not all `<image>` tags in the (Ad Hoc) Wikipedia XML collection correspond to images that are actually part of the Wikipedia image XML collection; images that are not part of this collection will not be visible to users during assessments. Therefore, we also removed all results that contained references to images that are not in the Wikipedia image XML collection. This way, we made sure all our returned fragments contain at least one *visible* image.

The results of our official submissions are presented in Table 4. Given our analysis of priors in Section 3.2, further experimentation is needed to determine whether other priors (e.g., quadratic and mixed priors) would lead to better performace. Finally, Table 4 also presents the results of our Ad Hoc Focused runs on the MMfragments topic subset, which indicate the usefulness of our filtering approach in the context of topics with clear multimedia character.

Table 4. Results for the CWI/UTwente official MMfragments Focused submissions and Ad Hoc Focused runs on the MMfragments topic subset. The table shows the rank of the run among official submissions, the run identifier and the interpolated precision at 0.01 recall.

rank	runID	iP[0.01]
1	article_MM	0.3389
4	star_loglength_MM	0.2467
5	star_lognormal_MM	0.0595
-	star_loglength	0.2325
-	star_lognormal	0.1045

Table 5. Results for the CWI/UTwente official submissions and additional runs to the MMimages task. The table shows the rank of the run among official submissions, the run identifier and Mean Average Precision.

rank	runID	MAP
1	title_MMim	0.2998
3	article_MMim	0.2240
5	figure_MMim	0.1551
-	title_MMim_lengthPrior	0.3094
-	title_MMim_logLengthPrior	0.3066

4.2 MMimages task

For MMimages, the aim is to retrieve documents (images + their metadata) from the Wikipedia image XML collection. Similarly to the Ad Hoc and MMfragments tasks, our submitted runs are based on the language modelling approach. Each image is represented either by its textual metadata in the Wikipedia image XML collection, or by its textual context when that image appears as part of a document in the (Ad Hoc) Wikipedia XML collection.

To be more specific, we submitted the following three runs:

title_MMim Create a stemmed index using the metadata accompanying the images in the Wikipedia image XML collection, and perform an article run using only the topics' title field: `//article[about(.,$title)]`.

article_MMim Rank the articles in the (Ad Hoc) Wikipedia XML collection using each topic's title field and retrieve the images that these articles contain. Filter the results, so that only images that are part of the Wikipedia image XML collection are returned.

figure_MMim Rank the figures with captions in the (Ad Hoc) Wikipedia XML collection using each topic's title field (`//figure[about(.,$title)]`) and return the images of these figures (ensuring that these images are part of the Wikipedia image XML collection).

Table 5 presents the Mean Average Precision (MAP) of these runs, whereas Figure 4 compares them against all the runs submitted to the MMimages task. Our experimental results indicate that these text-based runs give a highly competitive performance on the MMimages task.

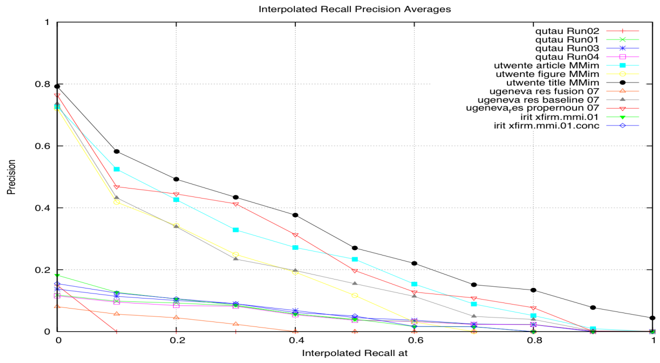


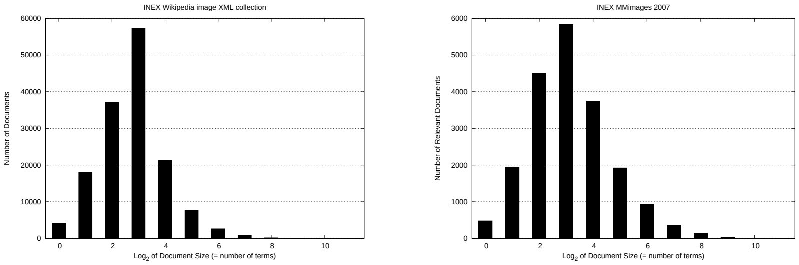
Fig. 4. MMimages results: CWI/Utwente runs compared to all submitted runs.

We incorporated a document prior based on length (defined as the number of terms in the metadata), **title_MMim_lengthPrior**, and the log of this length, **title_MMim_logLengthPrior**. By defining the priors in that manner, we are able to apply them without performing any training. Our results in Table 5 indicate that both priors improve over the corresponding baseline, with the length prior improving the most.

These runs are based on the assumption that the distribution of document size is different for relevant and non-relevant images. We perform a retrospective analysis of the distribution of length in the MMimages collection (Figure (5a)), and the relevant documents for 2007 (Figure (5b)). While the collection contains many small documents, these are rarely relevant. If we would not pay attention to document length and just use a retrieval model that does not have a bias for documents of any size, we would retrieve too many small documents. Simply giving a bias towards longer documents in the context of the INEX MMimages task has the potential of improving the retrieval result, which is confirmed by our evaluation experiments.

5 Entity Ranking by Relevance Propagation

We also participated in this year's entity ranking task. The queries here ask for a ranked list of entities, e.g., movies, flags, or diseases. Entities are usually identified by their name and type. An entity of type movie would be identified by its title. In general, the entity ranking task clearly differs from document ranking, since it requires to estimate the relevance of items that do not have text content [12, 15]. In this case, the ranking can only be done by propagating the relevance from retrieved text fragments to their contained entities. Using Wikipedia as the corpus for entity ranking experiments, the setting changes slightly. In order to use the existing mark-up of the corpus – instead of employing taggers for named



(a) Wikipedia image XML documents (b) MMimages 2007 relevant documents

Fig. 5. Size distribution of metadata in Wikipedia image XML collection and of metadata of images relevant to MMimages 2007 topics

entity recognition – the only entities considered were those that have their own Wikipedia article. An entity is contained in an article when it is linked by that article. In consequence, the distinction of articles and entities is abandoned. Since entities have their own article, they can also be ranked directly by their content.

In the context of Wikipedia, the type of an entity is defined by the categories assigned to the entity's article. An entity can thus have several types. Furthermore, Wikipedia categories are hierarchically organized. We can thus assume that an entity does not only belong the categories assigned to it, but also to ancestor categories. However, the hierarchy of Wikipedia's categories does not form a strict tree, and thus moving too far away from the original categories can lead to unexpected type assignments.

Our entity ranking approach can be summarized by the following processing steps: (1) initial retrieval of articles, (2) building of an entity graph, (3) relevance propagation within the graph, and (4) filtering articles by the requested type. The notion of *entity graph* stands here for a query-dependent link graph, consisting of all articles (entities) returned by the initial retrieval as vertices and the link-structure among them forming the edges. Links to other articles not returned in the initial ranking are not considered in the entity graph. The entity graph can later be used for the propagation of relevance to neighbouring nodes. Starting with web retrieval [10, 7, 13], graph based ranking techniques have been recently used in several fields of IR [3, 9, 2].

5.1 Baseline: Entity Retrieval by Description Ranking

The simplest and most obvious method for entity retrieval is the ranking of their textual descriptions with some classic document retrieval method. In our experiments, we rank Wikipedia articles representing entities using a language model based retrieval method:

$$P(Q|e) = \prod_{t \in Q} P(t|e), \tag{2}$$

$$P(t|e) = (1-\lambda_C)\frac{tf(t,e)}{|e|} + \lambda_C \frac{\sum_{e'} tf(t,e')}{\sum_{e'} |e'|} \quad (3)$$

where $tf(q,e)$ is a term frequency of q in the entity description e, $|e|$ is the description length and λ_C is a Jelinek-Mercer smoothing parameter - the probability of a term to be generated from the global language model. In all our experiments, λ_C is set to 0.8, which is standard in retrieval tasks.

However, due to several reasons this approach may produce unsatisfactory results. First, many entities have too short or empty descriptions, especially those that appear in novel and evolving domains that are just becoming known. Thus, many entities may get scores close to zero and not appear in the top. Second, many entities are described by showing the associations with other entities and in terms of other entities. This means that query terms have lesser chance in appearing in the content of a relevant description, since some concepts mentioned in its text are not explained because explanations can be found in their own descriptions.

5.2 Entity Retrieval Based on K-Step Random Walk

In our follow-up methods, we consider that relevance propagation from initially retrieved entities to the related ones is important. We imagine and model the process in which the user, after seeing initial list of retrieved entities:

- selects one document and reads its description,
- follows links connecting entities and reads descriptions of related entities.

Since we consider this random walk as finite, we assume that at some step a user finds the relevant entity and stops the search process. So, we iteratively calculate the probability that a random surfer will end up with a certain entity after K steps of walk started at one of the initially ranked entities. In order to emphasize the importance of entities to be in proximity to the most relevant ones according to the initial ranking, we consider that both (1) the probability to start the walk from certain entity and (2) the probability to stay at the entity node are equal to the probability of relevance of its description.

$$P_0(e) = P(Q|e) \quad (4)$$

$$P_i(e) = P(Q|e)P_{i-1}(e) + \sum_{e' \to e} (1 - P(Q|e'))P(e|e')P_{i-1}(e'), \quad (5)$$

The probabilities $P(e|e')$ are uniformly distributed among links outgoing from the same entity. Finally, we rank entities by their $P_K(e)$.

Linear Combination of Step Probabilities It is also possible to estimate entity relevance using several finite walks of different lengths at once. In the following modification of the above described method, we rank entities considering a weighted sum of probabilities to appear in the entity node at different steps:

$$P(e) = \mu_0 P_0(e) + (1-\mu_0)\sum_{i=1}^{K} \mu_i P_i(e) \quad (6)$$

In our experiments we set μ_0 to 0.5 and distribute $\mu_1 \ldots \mu_K$ uniformly.

5.3 Entity Retrieval Based on Infinite Random Walk

In our second approach, we assume that the walk in search for relevant entities consists of countless number of steps. The stationary probability of ending up in a certain entity is considered to be proportional to its relevance. Since the stationary distribution of a described discrete Markov process does not depend on the initial distribution over entities, the relevance flow becomes unfocused. The probability to appear in a certain entity node becomes dependent only on its centrality, but not on its closeness to the sources of relevance. To solve this issue, we introduce regular jumps to entity nodes from any node of the entity graph after which the walk restarts and the user follows inter-entity links again. We consider that the probability of jumping to a specific entity equals to the probability of relevance of its description. This makes a random walker visit entities which are situated closer to the initially highly ranked ones more often during normal walk steps. The following formula is used for iterations until convergence:

$$P_i(e) = \lambda_J P(Q|e) + (1 - \lambda_J) \sum_{e \to e'} P(e|e') P_{i-1}(e') \tag{7}$$

where λ_J is the probability that, at any step, the user decides to make a jump and not to follow outgoing links anymore. The described discrete Markov process is stochastic and irreducible, since each entity is reachable due to the introduced jumps, and hence has a stationary distribution. Consequently, we rank entities by their stationary probabilities $P_\infty(e)$.

5.4 Experiments

We trained our models using the 28 queries from the Ad-Hoc XML Retrieval task that are also suitable for the entity ranking task. All our algorithms start from the retrieval of articles from the collection using a language modelling approach to IR for scoring documents. Then, we extract entities mentioned in these articles and build entity graphs. For the initial article retrieval, as well as for the graph generation, the PF/Tijah retrieval system was employed. For this experiment, we generated XQueries that directly produce entity graphs in *graphml* format given a title-only query. We tuned our parameters by maximization of the MAP measure for 100 initially retrieved articles.

For the following methods, we discuss their performance first on the training and then on the test data:

- **Baseline**: the baseline method which ranks entities by the relevance of their Wikipedia-articles (see Equations 2, 3),
- **K-Step RW**: the K-step Random Walk method which uses multi-step relevance propagation with K steps (see Equations 4, 5),
- **K-Step RWLin**: the K-step Random Walk method which uses the linear combination of entity relevance probabilities at different steps up to K (see Equation 6),

– **IRW**: the Infinite Random Walk method which ranks entities by the probability of reaching them in infinity during non-stop walks (see Equation 7).

For the Entity Retrieval task, we have a query and the list of entity categories as input. However, according to the track guidelines and our own intuition, relevant entities could be found to be out of the scope of given categories. Preliminary experiments have shown that using parent categories of any level spoiled the performance of the Baseline method. However, it was very important to include child categories up to 3rd level both for our Baseline method and for our remaining methods which require tuned parameters (see Figure 6). This probably means that queries were created with an assumption that given categories should be the greatest common super-types for the relevant entities. It must be mentioned that we used entities of all categories for the graph construction and relevance propagation, and filtered out entities using the list of allowed categories only at the stage of result output.

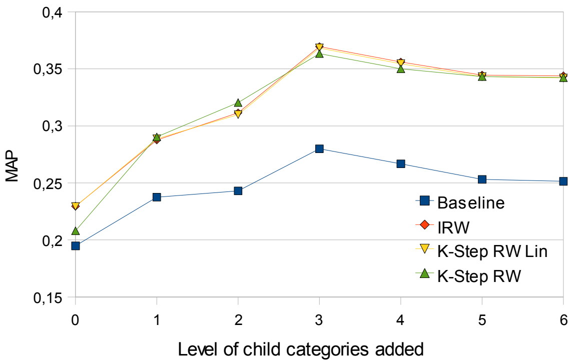


Fig. 6. MAP performance of all methods for different levels of child categories added

In all our methods, except the Baseline, we had to tune one specific parameter. For the K-step RW and K-step RWLin methods, we experimented with the number of walking steps. As we see in Figure 7, both methods reach their maximum performance after 3 steps only. The K-step RW Lin method seems to be more robust to the parameter setup. This probably happens because it smooths the probability to appear in a certain entity after K steps, with probabilities of visiting it earlier. The rapid decrease of performance for even steps for the K-step RW method can be explained in the following way. A lot of relevant entities are only mentioned in the top ranked entity descriptions and do not have their own descriptions in this top, due to their low relevance probability or due to their absence in the collection. The relevance probability of these “outsider” entities entirely depends on the relevance of related entities, which are not relevant entities themselves (for example, do not match the requested entity type), but tell a lot about the ranked entity. So, all “outsider” entities have direct (backward) links only to the entities with descriptions in the top.

Since we always start walking only from the latter entities, the probability to appear in "outsider" entities at every even step is close to zero.

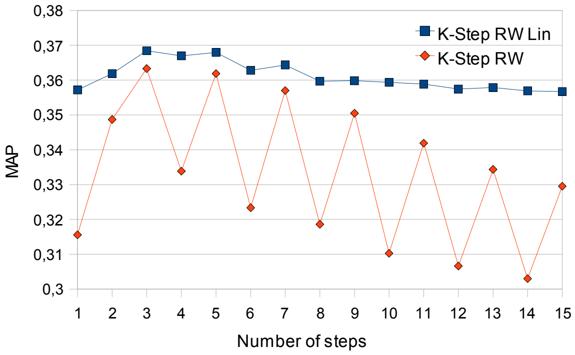


Fig. 7. MAP performance for two methods and different numbers of steps

We also experimented with the probability to restart the walk from initially ranked entities for the IRW method. According to results shown in Figure 8, values between 0.3 and 0.5 seem to be optimal. This actually means that making only 2-3 steps (before the next restart) is the best strategy. which is also the case for the finite random walk methods.

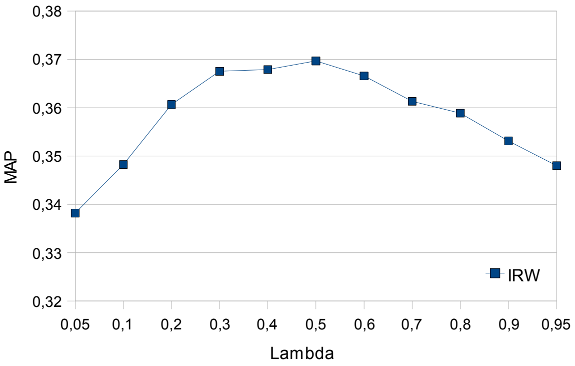


Fig. 8. MAP performance of IRW method for different values of jumping probability

To sum the things up, our experiments with the training data showed that all our three methods significantly outperform the **Baseline** method. However, the **K-Step RW** method produced a bit worse results than the other two.

As we see in Table 6, our final results on the test data show that both the **K-Step RWLin** and the **IRW** methods are equally more effective than the

runID	MAP
Baseline	0.291
K-Step RW	0.281
K-Step RWLin	0.306
IRW	0.301

Table 6. Final results for all methods for the Entity Ranking task

Baseline method. The fact that the **K-Step RW** could not outperform the **Baseline** method in our final experiments confirms its lower robustness with respect to the proper parameter setup.

6 Conclusions

This is the second year that CWI and University of Twente used PF/Tijah in INEX. The flexibility of this system is clearly demonstrated through its application in INEX tracks as diverse as ad hoc structured document retrieval, retrieval of multimedia documents and document fragments, and entity ranking.

The unigram language modelling approach we have previously applied in Ad Hoc element retrieval tasks retrieves short elements. Given that our analysis of last year's results indicates that the relevant elements tend to be longer than the ones our approach retrieves, the incorporation of length priors would be beneficial. For the Focused subtask, further experimentation is needed to determine whether the priors indicated by our recent analysis would yield better performance, whereas for the Best in Context and Relevant in Context subtasks, we need to examine in more detail our filtering strategies.

Our text only approach to Multimedia retrieval was very successful on the MMimages task. Further experimentation on the MMfragments task would reveal whether more appropriate filtering techniques or alternative priors would improve our results.

The experiments with our approaches for entity ranking demonstrated the advantage of multi-step relevance propagation from textual descriptions to related entities over the simple ranking of entity textual descriptions. The further improvement seems especially challenging because all our three methods showed quite similar effectiveness.

7 Acknowledgements

This material is based on work funded in part by the European Union via the European Commission project VITALAS (contract no. 045389), the Dutch National project MultimediaN, and the Netherlands Organisation for Scientific Research (NWO).

References

1. P. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner. MonetDB/XQuery: A fast XQuery processor powered by a relational engine. In *Proceedings of the 25th ACM SIGMOD International Conference on Management of Data*, pages 479–490, 2006.
2. P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *Proceedings of the 14th ACM CIKM International Conference on Information and Knowledge Management*, pages 373–380, 2005.
3. N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of the 30th ACM SIGIR Annual International Conference on Research and Development in Information Retrieval*, pages 239–246, 2007.
4. D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 513 of *Lecture Notes in Computer Science*, pages 569–584. Springer-Verlag, 1998.
5. D. Hiemstra, H. Rode, R. van Os, and J. Flokstra. PF/Tijah: text search in an XML database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR)*, 2006.
6. J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In *Proceedings of the 27th ACM SIGIR Annual International Conference on Research and Development in Information Retrieval*, pages 80–87, 2004.
7. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
8. W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM SIGIR Annual International Conference on Research and Development in Information Retrieval*, pages 27–34, 2002.
9. A. Kritikopoulos, M. Sideri, and I. Varlamis. Blogrank: ranking weblogs based on connectivity and similarity features. In *Proceedings of the 2nd International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications*, page 8, 2006.
10. P. Lawrence, B. Sergey, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
11. J. List, V.Mihajlovic, G.Ramirez, A. de Vries, D. Hiemstra, and H. Blok. Tijah: Embracing IR methods in XMl databases. *Information Retrieval*, 8(4):547 – 570, 2005.
12. P. Serdyukov, H. Rode, and D. Hiemstra. University of Twente at the TREC 2007 Enterprise Track: Modeling relevance propagation for the expert search task. In *Proceedings of the 16th Text REtrieval Conference (TREC 2007)*, 2007.
13. A. Shakery and C. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *Proceedings of the 15th ACM CIKM International Conference on Information and Knowledge Management*, pages 550–558, 2006.
14. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th ACM SIGIR Annual International Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.
15. H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *Proceedings of the 16th ACM CIKM International Conference on Information and Knowledge Management*, pages 1015–1018, Lisbon, Portugal, 2007.