DCU and ISI@INEX 2010: Ad-hoc and Data-Centric tracks

Debasis Ganguly[1], Johannes Leveling[1], Gareth J. F. Jones[1]
Sauparna Palchowdhury[2], Sukomal Pal[2], and Mandar Mitra[2]

[1]CNGL, School of Computing, Dublin City University, Dublin, Ireland
[2]CVPR Unit, Indian Statistical Institute, Kolkata, India
{dganguly, jleveling, gjones}@computing.dcu.ie
sauparna.palchowdhury@gmail.com {sukomal_r, mandar}@isical.ac.in

Abstract. We describe the participation of Dublin City University (DCU) and the Indian Statistical Institute (ISI) in INEX 2010. The main contributions of this paper are: i) a simplified version of Hierarchical Language Model (HLM) which involves scoring XML elements with a combined probability of generating the given query from itself and the top level article node, is shown to outperform the baselines of Language Model (LM) and Vector Space Model (VSM) scoring of XML elements; ii) the Expectation Maximization (EM) feedback in LM is shown to be the most effective on the domain specific collection of IMDB; iii) automated removal of sentences indicating aspects of irrelevance from the narratives of INEX ad-hoc topics is shown to improve retrieval effectiveness.

1 Introduction

Traditional Information Retrieval (IR) systems return whole documents in response to queries, but the challenge in XML retrieval is to return the most relevant parts of XML documents which meet the given information need. Since INEX 2007 [1] arbitrary passages are also permitted as retrievable units, besides XML elements. A retrieved passage can be a sequence of textual content either from within an element or spanning a range of elements. INEX 2010 saw the introduction of the restricted version of the "Focused" task which is designed particularly for displaying results on a mobile device with limited screen size. The Ad-hoc track tasks comprises of the following tasks: a) the "Restricted Focused" task which asks systems to return a ranked list of elements or passages to the user restricted to at most 1000 characters per topic; b) the (un)restricted "Relevant in Context" tasks which ask systems to return relevant elements or passages grouped by article, a limit of at most 500 characters being imposed on the restricted version; and c) the "Efficiency" task which aims at retrieval in an efficient manner allowing systems to return thorough article level runs. We also participated in the new "Data Centric track" which is similar to Ad-hoc retrieval of elements or passages on a domain specific collection of IMDB movie pages. In INEX-2010 we submitted article level runs for the Efficiency task and element level runs for Restricted Focused and (Un)Restricted Relevant-In-Context tasks.

In addition we submitted both article and element level runs to the Data Centric track.

The remainder of this paper is organized as follows: Section 2 elaborates on the approaches to indexing and retrieval of whole documents followed by Section 3, which describes the strategy for measuring the similarities of the individual XML elements to the query. In Section 4 we propose a simplified version of HLM for XML retrieval, which involves scoring an XML element with a linear combination of the probability of generating the given query from itself and its root article. Using the INEX-2009 topic set for training and 2010 topic set for testing, we show that it outperforms the standard LM and VSM methods for scoring XML elements. Section 5 explores the effectiveness of Blind Relevance Feedback (BRF) on the domain specific collection of IMDB movie database. Section 6 describes post-official submission research analyzing the narrative parts of INEX topics and automatically filtering out the sentences indicating a negative impact on relevance to improve retrieval effectiveness. Section 7 concludes the paper with directions for future work.

2 Document Retrieval

2.1 Preprocessing

Similar to INEX 2009, for extracting useful parts of documents, we shortlisted about thirty tags that contain useful information: <*p*>, <*p1*>, <*st*>, <*section*> <*ip1*>, <*it*>, , <*fnm*>, <*snm*>, <*atl*>, <*ti*>, <*h2a*>,<*h*>, <*wikipedialink*>, , <*outsidelink*>, <*td*>, <*body*> etc. [2]. Documents were parsed using the libxml2 [1] parser, and only the textual portions included within the selected tags were used for indexing. Similarly, for the topics, we considered only the *title* and *description* fields for indexing, and discarded the *inex-topic*, *castitle* and *narrative* tags. No structural information from the queries was used.

The extracted portions of the documents were indexed using single terms and a pre-defined set of 100,000 most frequent phrases (extracted by the N-gram Statistics Package [2](NSP) on the English Wikipedia text corpus), employing the SMART[3] system. Words in the standard stop-word list included within SMART were removed from both documents and queries. The default stemmer implementation of SMART which is a variation of the Lovin's stemmer [3] was used.

2.2 Language Model (LM) Term Weighting

Our retrieval method is based on the Language Model (LM) approach proposed by Hiemstra [4]. In this subsection we summarize the LM method for IR used for document retrieval in this work. In LM based IR, a document d is ranked by

[1] `http://xmlsoft.org/`
[2] http://www.d.umn.edu/~tpederse/nsp.html
[3] `ftp://ftp.cs.cornell.edu/pub/smart/`

a linear combination of estimated probabilities $P(q|d)$ of generating a query q from the document d and $P(t_i)$ of generating the term from the collection. The document is modelled to choose $q = \{t_1, t_2 \dots t_n\}$ as a sequence of independent words as proposed by Hiemstra [4].

$$P(q|d) = P(d) \prod_{i=1}^{n} \lambda_i P(t_i|d) + (1 - \lambda_i) P(t_i) \tag{1}$$

$$\log P(q|d) = \log P(d) + \sum_{i=1}^{n} \log(1 + \frac{\lambda_i}{1 - \lambda_i} \frac{P(t_i|d)}{P(t_i)}) \tag{2}$$

$P(d)$ is the prior probability of relevance of a document d and it is typically assumed that longer documents have higher probability of relevance. The term weighting equation can be derived from Equation 1 by dividing it with $(1 - \lambda_i)P(t_i)$ and taking logarithm on both sides to convert the product to summation. This transformation also ensures that the computed similarities between documents and a given query are always positive. We index each query vector $\boldsymbol{q}$ as $q_k = tf(t_k)$ and each document vector $\boldsymbol{d}$ as $d_k = \log(1 + \frac{P(t_k|d)}{P(t_k)} \frac{\lambda_k}{1-\lambda_k})$, so that the dot product $\boldsymbol{d} \cdot \boldsymbol{q}$ gives the likelihood of generating $\boldsymbol{q}$ from $\boldsymbol{d}$ and hence can be used as the similarity score to rank the documents.

3 XML Element Retrieval

For the element-level retrieval, we adopted a 2-pass strategy. In the first pass, we retrieved 1500 documents for each query using the LM retrieval method as described in the previous section 2.2. In the second pass, these documents were parsed using the libxml2 parser, and leaf nodes having textual content were identified. The total set of such leaf-level textual elements obtained from the 1500 top-ranked documents were then indexed and compared to the query as before to obtain the final list of 1500 retrieved elements. The preprocessing steps are similar to those as described in section 2.1. The following section provides details of our proposed method of scoring XML elements.

3.1 Simplified Hierarchical Language Model

Motivation The objective of the focused task is to retrieve short relevant chunks of information. Scoring an XML element by its similarity with the query may retrieve short XML elements such as a small paragraph or sub-sections with a dense distribution of query terms in the top ranks. However, there is an implicit risk associated with the retrieval of short high scoring elements, namely that the text described in the short element might be a digression from the main topic of the root article. Thus it is unlikely that the retrieved text from the XML element would serve any useful purpose to the searcher because it would not have the necessary context information to do so. As an example consider the artificial paragraph of text as shown in Fig. 1. Now let us imagine that a searcher's query

We crawled the IMDB movie collection and categorized the crawled IMDB data into categories such as movies, actors etc. Movie reviews and ratings were also stored.

Fig. 1: An artificial paragraph of text to illustrate the implicit risk of out-of-context retrieval with response to a query "top-10 IMDB movies actors".

is "top-10 IMDB movies actors". Let us also imagine that this paper has been converted into an XML formatted document and put into the INEX document collection. The contents of the above paragraph thus could be retrieved at a top rank because of its high similarity due to the presence of three matching query terms as highlighted with boxes. However, the searcher was certainly not looking for a technical paragraph on indexing the IMDB collection. This shows that the retrieval model for XML elements needs to be extended to take into consideration the distribution of query terms in the root articles in addition to the element itself. If the query terms are too sparse in the root article, then it is more likely that the XML element itself is an out-of-context paragraph, e.g. the highlighted words in Fig. 1 are very sparsely distributed throughout the rest of this document, which suggests that the shown artificial paragraph shown is not a good candidate for retrieval.

It is also desirable to retrieve an element with a few query terms if the other missing terms are abundant in the article. This is particularly helpful for assigning high scores to elements (sections or paragraphs) which densely discuss a single sub-topic (the sub-topic typically being one specific facet of the user information need), whereas the rest of the article throws light on other general facets hence providing the necessary context for the specific subtopic.

The Scoring Function An extension of LM for Field Search named Field Language Model (FLM) was proposed by Hiemstra [4] which involves scoring a document according to the probability of generation of the query terms either from the document itself as a whole, or from a particular field of it (e.g title) or from the collection. We propose to assign a score to the constituent element itself from the root article evidence thus differing from FLM in the directionality of assignment of the scores. We use Equation 3 to score an XML element e.

$$P(q|e) = P(e) \prod_{i=1}^{n} \{\mu_i P(t_i|e) + \lambda_i P(t_i|d) + (1 - \lambda_i - \mu_i) P(t_i)\} \tag{3}$$

In Equation 3 λ_i denotes the probability of choosing the query term t_i from d (the root article of the element e), whereas μ_i denotes the probability of choosing t_i from the element text. The residual event involves choosing t_i from the collection. Thus even if a query term t_i is not present in the element a non zero probability of generation is contributed to the product. Two levels of smoothing are employed in this case.

Sigurbjrnsson et. al. compute the article score and element scores separately by Equation 2 and then use a linear combination to capture the context of an element [5]. Their method has three parameters λ_{elt} and λ_{art} for the element and whole article scores respectively, and another α for combining these two. In contrast, we extend the element scoring equation itself, thus leading to a more tight coupling with the root article score and avoiding one extra parameter. Ogilvie and Callan developed the general HLM which involves two way propagation of the LM element scores from the root to the individual leaf nodes and vice-versa [6]. Our model is much simpler in the sense that we use only the current node and the top level article node for the score computation. We call this method Simplified Hierarchical Language Model (SHLM) because we restrict our smoothing choice to the root article element only in addition to the collection. The SHLM equation can be further simplified by using $\lambda_i = \lambda \wedge \mu_i = \mu$ $\forall i = 1 \ldots n$. Experimental evaluation of SHLM for ad-hoc XML retrieval is provided in section 4. It can be seen that Equation 3 addresses the motivational requirements as described in the previous section in the following ways:

a) An element e_1 which has a query term t only in itself but not anywhere else in the top level article, scores lower than an element e_2 which has the term present both in itself and somewhere else in the article. Thus the model favours elements with some pre-defined contextual information about the query terms over individual snippets of information which do not have any associated context.
b) An element with a few of the given query terms might be retrieved at a high rank if the missing terms are abundant in the article.

4 Ad-hoc Track Experiments and Results

We trained our system using the INEX 2009 topic set. All the initial article level runs were performed using LM retrieved as described in Equation 1. We assign $\lambda_i = \lambda \quad \forall i = 1 \ldots n$ and also assigned uniform prior probabilities to the documents. After a series of experiments by varying λ we found that best retrieval results are obtained with $\lambda = 0.4$ and henceforth we use this setting for all the article level LM runs reported in the paper. We officially submitted three article level runs containing 15, 150 and 1500 documents as *thorough* runs. We conducted a post-hoc analysis after the INEX results were officially released. This revealed that there was a bug in our retrieval control-flow where we used inverted lists constituted from the raw document vectors instead of the LM weighted ones. The results for the thorough runs, along with post-submission corrected versions are shown in Table 1.

We submitted 3 element level runs for the restricted focused task. The first two subsections report the training of SHLM on the INEX 2009 topics and the last section reports the official submissions under the restricted focused task.

Tuning SHLM To test the effectiveness of SHLM (as outlined in section 3.1) for element retrieval, we run SHLM with different combinations of λ and μ

Table 1: Official evaluation of the thorough runs

Run Id	# docs retrieved	Submitted		Corrected	
		MAP	MAiP	MAP	MAiP
ISI2010_thorough.1500	1500	0.0431	0.0846	**0.1539**	**0.1750**
ISI2010_thorough.150	150	0.0309	0.0826	0.1185	0.1421
ISI2010_thorough.15	15	0.0110	0.0714	0.0647	0.0930

(simplifying Equation 3 by employing $\lambda_i = \lambda \wedge \mu_i = \mu \ \forall i = 1 \ldots n$) on the INEX 2009 training topics. A revisit of Equation 3 suggests that a higher value of μ as compared to λ attaches too much importance to the presence of the query terms. While this might be good for queries with highly correlated terms, typically user queries are faceted, each term representing one such facet. It is highly unlikely that a single section or paragraph would cover all the facets. The more likely situation is that a small paragraph would cover one facet of the user's information need whereas the other facets are covered somewhere else in the document. Our hypothesis is that a value of μ lower than λ ensures that we lay emphasis on not retrieving out-of-context small elements. In this case we expect better retrieval performance by setting $\mu < \lambda$.

Another critical aspect to explore in the model of Equation 3 is the issue of assigning prior probabilities to the elements. Singhal [7] analyzes the likelihood of relevance against the length of TREC documents and reports that longer documents have a higher probability of relevance. While this scheme of assigning document prior probabilities proportional to their lengths suits the traditional ad-hoc retrieval of documents (the retrieval units being whole documents) from the news genre, for a more flexible retrieval scenario such as the Restricted Focused INEX task where retrieval units can be arbitrary passages and shorter passages are favoured over longer ones, it might be worth trying to assign prior probabilities to elements inversely proportional to their lengths.

SHLM Results As our baseline we use standard LM scoring of the elements which is a special case of SHLM obtained by setting $\lambda = 0$. To verify our hypothesis that λ should be higher than μ, we ran two versions of SHLM one with $\lambda < \mu$ and the other $\mu > \lambda$. Table 2 reports the measured retrieval effectiveness of the different cases and also shows the effect on precision for the three different modes of element priors - i) uniform, ii) proportional and iii) inversely proportional for the case $\mu < \lambda$. Table 2 provides empirical evidence to support the hypothesis that elements when scored with contextual information from their root article yield better retrieval results. The first row of the table reports the case where elements are LM weighted without any root article information. It can be seen that the first row yields the least $iP[0.01]$ value. The table also justifies the hypothesis of assigning $\mu < \lambda$ since $iP[0.01]$ of the third and fifth rows are higher than that of the second row.

Table 2: SHLM for element retrieval for INEX 2009 topics

λ	μ	Element Prior probability	Retrieval Effectiveness iP[0.01]	iP[0.05]	iP[0.10]	MAiP
0.0	0.15	Uniform	0.2639	0.1863	0.1335	0.0448
0.15	0.25	Uniform	0.4082	0.2648	0.1894	0.0566
0.25	0.15	Uniform	**0.5256**	**0.3595**	0.2700	0.0991
0.25	0.15	Shorter favored	0.3459	0.1682	0.0901	0.0314
0.25	0.15	Longer favored	0.4424	0.3582	**0.2787**	**0.1064**

Official Results The restricted focused task required systems to return a ranked list of elements or passages restricted to at most 1000 characters per topic. The evaluation metric used was P@500 characters. Since this metric favours retrieval runs with a high precision at low recall levels (recall is expected to be low when only 1000 characters are retrieved), we use the settings as reported in the third row of Table 2 i.e. with the settings $(\lambda, \mu) = (0.25, 0.15)$ with uniform element prior probability. We perform SHLM element retrieval on i) our best performing LM retrieved article level run ($\lambda = 0.4$), and ii) reference BM25 run provided by the INEX organizers. To show that SHLM performs better than the pivoted length normalized scoring which was our element level retrieval strategy for INEX 2009 [2], we also submitted a run scoring the elements by Equation 4.

$$normalization = 1 + \frac{slope}{(1 - slope)} \cdot \frac{\#unique\ terms}{pivot} \tag{4}$$

Table 3 shows that the corrected SHLM based element retrieval on the reference run yields the highest character-precision among our runs. We also see that

Table 3: Official evaluation of the Focused runs

Run Id	Methodology	P@500 chars submitted	corrected
ISI2010_rfcs_ref	SHLM element retrieval on article level reference run	0.2451	**0.3755**
ISI2010_rfcs_flm	SHLM element retrieval ($\mu < \lambda$ and uniform prior of the elements) on article level LM run	0.2151	0.2841
ISI2010_rfcs_vsm	Pivoted normalized element retrieval on article level LM run	0.1289	0.2343
LIP6-OWPCparentFo (Best run)		**0.4125**	

the best character precision we achieved ranks third within the list of official submissions (after LIP6-OWPCparentFo and DURF10SIXF).

SHLM clearly outperforms pivoted normalized element retrieval on the same document level run showing that given the same document level run, it is more effective than VSM based element retrieval. Table 3 also shows that SHLM outputs a better element level run for a better input article run as evident from the first and second rows (MAP of the reference run is higher than our LM based article run).

5 Data Centric Track Experiments and Results

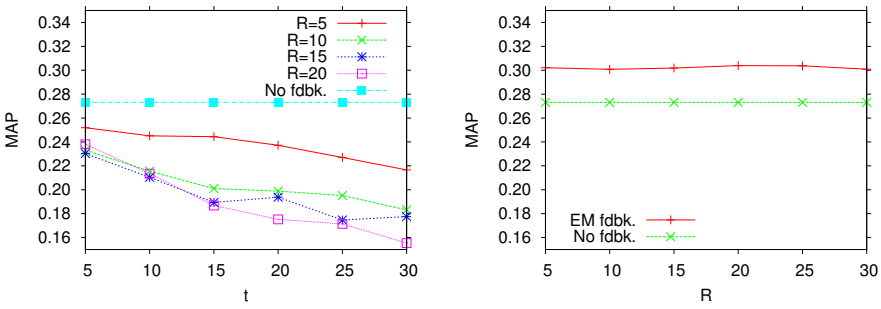
We indexed the IMDB collection using SMART in a similar way as outlined in section 2, the only difference being that we did not use a pre-extracted list of commonly occurring bigrams for constructing the phrase index. Our Data Centric official submissions also suffered from the same bug as with our Ad-hoc submissions. In this section we report a set of refined results after carrying out experiments with a corrected version.

Our approach in this track comprises of exploring the standard IR techniques on a domain specific collection like the movie database. Our initial experiments show that we get the optimal baseline MAP by using $\lambda = 0.6$. While performing feedback experiments we found that a selection of query expansion terms by the LM score [8] results in worse retrieval performance. Fig. 2a shows the effect on retrieval performance (MAP) for query expansion with different settings of R (the number of pseudo-relevant documents used) and t (the number of terms used for query expansion). We implemented the EM feedback in SMART as proposed by Hiemstra [4] where each λ_i, associated with the query term t_i, is modified aiming to maximize the expectation of the LM term generation probability from the top ranked pseudo-relevant documents as shown in Equation 5.

$$\lambda_i^{p+1} = \frac{1}{R} \sum_{j=1}^{R} \frac{\lambda_i^p P(t_i|D_j)}{\lambda_i^{(p)} P(t_i|D_j) + (1 - \lambda_i^{(p)}) P(t_i)} \tag{5}$$

We use only one iteration of the feedback step i.e. we calculate λ_i^1s from the initial $\lambda_i^0 = \lambda$ for each i. We also tried out applying EM to an expanded query with additional terms but we found out that it did not improve the MAP. The results as shown in Figure 2b reveal that EM performs better than the LM term based expansion as shown in Figure 2a. While doing a per-topic analysis of the document retrieval for the IMDB collection, we made the following interesting observation. Query 2010015 reads "May the force be with you" of which all are stopwords except the word *force*. As a result the obtained MAP for this query is 0.

A characteristic of the IMDB collection is that the documents are grouped into categories such as *movies*, *actors* etc. To find out if relevance is biased towards one of the categories, we computed the percentage of relevant documents in each of the categories from the article level manual assessments. We also computed the percentage of retrieved documents in each of category.



(a) LM score based query expansion (b) EM

Fig. 2: Feedback effects on the IMDB collection

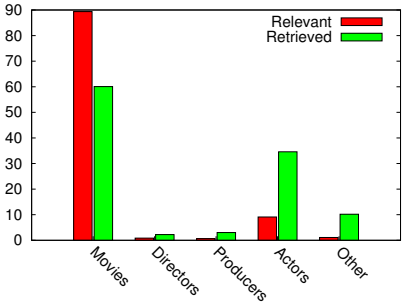


Fig. 3: Percentage of relevant and retrieved documents in the different categories

Fig. 3 shows that relevance is heavily biased to the movie documents suggesting that the searchers mostly seek movie pages in response to a submitted query. For the retrieved set we find that movie pages are retrieved highest in number, followed by actor pages and so on. The relative ranks of the number of hits for the relevant and retrieved categories are almost the same with an exception for the categories of producers and directors, where producer pages have the least number of hits in the relevant set, whereas this category is not with least number of hits for the retrieved set. Although relative ranks are similar, there is a noticeable difference in the percentages between the two sets, especially in the categories movies and actors which suggests that adjusting the prior relevance $P(d)$ (Equation 1) not according to the length of a document but according to its category could be a way to improve on the retrieval effectiveness. This would help to reduce the percentage gaps in the relevant and retrieved sets.

6 Query Processing Experiments and Results

Motivation It is easier for a user to formulate a complex information need in natural language rather than using terse keyword queries. The INEX topics have a detailed narrative (N) section that reinforces the information expressed in the title (T) and description (D) fields. Notably, in the narrative section, the topic creator specifies what he is not looking for.

One such INEX query is shown in Table 4. The emphasized sentence of N is the negative information.

To support our claim that such negative information can introduce a query drift towards non-relevant documents, we report a comparison of retrieval results, using queries processed using manual and automatic methods. Results show that the modified queries, with negation removed, yield higher retrieval effectiveness.

Table 4: An example INEX 2009 topic with negative information.

<topic id="2009080" ct_no="268">
<title>international game show formats</title>
<description>I want to know about all the game show formats that have adaptations in different countries.</description>
<narrative> Any content describing game show formats with international adaptations are relevant. *National game shows and articles about the players and producers are not interesting.*
</narrative>
<topic>

Approach Our unmodified set of queries is Q. From Q we create two new sets. The first one, P, consists of only those queries in Q which have negation, and, with these negative sentences or phrases removed. (P, stands for 'positive' in the context of this experiment). The second set, P_M ('positive', 'machine'-processed set), is similar, but negation was now automatically identified using a Maximum Entropy Classifier (Stanford Classifier) [4], and removed. In Table 6 the cardinalities of P and P_M differ because the classifier does not identify all the negative phrases and sentences with full accuracy. Some queries in Q, which have negation, and can be found in P, may not make it to P_M. We did retrieval runs using Q, P and P_M and noted the change in MAP (Refer to [9] for more details). The classifier performed well with accuracies around 90% (Table 5).

Retrieval Results We used the SMART retrieval system to index the document collection using pivoted document length normalized VSM (Equation 4), and the initial retrieval run was followed by a BRF run employing Rocchio feedback. For feedback we used the most frequent 20 terms and 5 phrases occurring in the top 20 pseudo-relevant documents setting $(\alpha, \beta, \gamma) = (4, 4, 2)$. Table 6 shows that the positive sets give an improvement in all cases.

Of particular interest is the P_M results; the P results are included only to refer to the maximum relative gain in performance that is achievable. Although, as expected, the relative gains for the P_M set is lower as compared to the P set, the differences between the two relative gains are not too big, which shows that the automated process does well.

The Wilcoxon test verifies that the differences in the relative gains of Q and P are statistically significant, corroborating the fact that removal of negation

[4] `http://nlp.stanford.edu/software/classifier.shtml`

Table 5: Classifier performance.

Test set	Training set	# of training sentences	Accuracy
2008	2007	589	90.4%
2009	2008	679	89.1%
2010	2009	516	93.8%

Table 6: Comparison of performance of the manually processed (P) and automatically processed (P_M) positive query sets.

Topic	Manually processed				Automatically processed			
Set	$\|P\|$	MAP_Q	MAP_P	change	$\|P_M\|$	MAP_Q	MAP_{P_M}	change
INEX 2008	44	0.2706	0.2818	4.1%	31	0.2638	0.2748	4.2%
INEX 2009	36	0.2424	0.2561	5.7%	30	0.2573	0.2581	0.3%
INEX 2010	26	0.3025	0.3204	6.0%	20	0.2922	0.2983	2.1%

improves performance. Also, a test of significance between P and P_M shows that their difference is statistically insignificant showing that the automated process is as good as the manual one.

One must note that our baseline comprises retrieval runs over the set Q where the queries are of maximum length. The queries in P and P_M, are shorter. It is expected that the retrieval effectiveness will improve with an increase in query size for the bag-of-words approach. We needed this to be empirically verified to rule out the possibility of an improvement in the retrieval effectiveness due to query length shortening.

Three retrieval runs were done using T, TD and TDN. The results in Table 7 show that there is a positive correlation between the MAP and query length. This eliminates the possibility of an improvement in MAP due to a negative correlation between query length and MAP for INEX topics. We also observe that the results for the 2009 set degrades across T, TD and TDN.

7 Conclusions and Future work

Through our participation in the ad-hoc track of INEX 2010, we revisited the LM element scoring strategy with the context information from parent articles. Our model is simpler as compared to [6] and has one less parameter as compared to [5]. Trial experiments on INEX 2009 topics show that it outperforms the baseline LM element retrieval of the elements. Official restricted focused runs show that SHLM element retrieval outperforms the pivoted normalized VSM element retrieval. Also our corrected official focused run ranks third among the submitted runs. The concept of SHLM can be extended to arbitrary passages by defining a series of fixed length window-subwindow structures.

For the data-centric track, we have shown that LM retrieval works well on a domain specific collection such as the movie database. We also show that

Table 7: MAP values for retrieval using increasing query size.

INEX year	T	TD	Δ (%)	TDN	Δ' (%)	trend (Δ, $\Delta' \geq 5\%$)
2008	0.2756	0.2815	2.14	0.2998	8.78	$- \uparrow$
2009	0.2613	0.2612	-0.03	0.2547	-2.52	$- -$
2010	0.2408	0.2406	-0.08	0.2641	9.67	$- \uparrow$

query expansion using terms selected by LM scores do not improve retrieval effectiveness whereas the EM feedback does well on this collection. We report a bias of relevance towards the movie and actor categories which suggests a possible future work of assigning higher prior probabilities of relevance for documents in these categories to help improve MAP.

Using the ad-hoc track topics, we show that it is possible to automate the process of removing negative sentences and phrases and that this improves retrieval effectiveness. Future work may involve detection of sub-sentence level negation patterns and handling complex negation phrases so as to prevent loss of keywords.

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) project.

References

1. Kamps, J., Geva, S., Trotman, A., Woodley, A., Koolen, M.: Overview of the INEX 2008 ad hoc track. In: Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008. (2008) 1–28
2. Pal, S., Mitra, M., Ganguly, D.: Parameter tuning in pivoted normalization for XML retrieval: ISI@INEX09 adhoc focused task. In: INEX. (2009) 112–121
3. Lovins, J.B.: Development of a stemming algorithm. Mechanical Translation and Computational Linguistics **11** (1968) 22–31
4. Hiemstra, D.: Using language models for information retrieval. PhD thesis, University of Twente (2001)
5. Sigurbjrnsson, B., Kamps, J., de Rijke, M.: An element-based approach to xml retrieval. In: IN: INEX 2003 Workshop Proceedings. (2004)
6. Ogilvie, P., Callan, J.: Hierarchical language models for XML component retrieval. In: INEX. (2004) 224–237
7. Singhal, A.: Term Weighting Revisited. PhD thesis, Cornell University (1996)
8. Ponte, J.M.: A language modeling approach to information retrieval. PhD thesis, University of Massachusetts (1998)
9. Palchowdhury, S., Pal, S., Mitra, M.: Using negative information in search. In: Proc. 2011 Second International Conference on Emerging Applications of Information Technology (EAIT 2011). (2011) 53–56