# UPF at INEX 2010
# Towards Query-type based Focused Retrieval

Georgina Ramírez

Universitat Pompeu Fabra, Barcelona, Spain
georgina.ramirez@upf.edu

**Abstract.** This paper describes our participation at INEX 2010. We participated in two different tracks: ad-hoc and data-centric. We first propose a classification of INEX topics and analyze several characteristics of the relevance assessments from INEX 2009 for each of the topic classes. The goal of our study is to investigate whether there are differences in relevance judgements between topic classes in order to use this information at retrieval time. We also present the experiments we performed on the INEX 2010 data. In the ad-hoc track we study the performance effects of changing the article order (fetching phase) while in the data-centric track we experiment with the use of different indices and retrievable element types. Our main finding is that indexing uniquely movie documents leads to much better performance than indexing the complete collection.

**Keywords:** XML, focused retrieval, INEX, query-based retrieval, query classification

## 1 Introduction

Retrieval tasks such as XML retrieval, where focused access to relevant information is provided, allow users to perform very focused searches and pose restrictions on the type of information being requested (e.g., I want references or experimental results). This is one of the reasons why several query languages and interfaces have been designed—to allow users to explicitly express more complex needs. However, these tools are not always available and users often specify in their *keyword* queries not only what they are looking for but also the type and the specificity of the information they are searching for. Thus, the number and variety of topic types that XML retrieval introduces differ from those of traditional document retrieval, where the task is to return whole documents.

In this paper we propose a classification of XML retrieval topics based on three different dimensions: 1) the type of information sought (general or restricted), 2) the specificity of the topic (generic or specific), and 3) the complexity of the topic (simple or compound). Once a classification is defined, it can be used in multiple ways. Our goal is to use it to perform specific retrieval strategies for each of the topic classes. A classification can also help to provide

a more balanced benchmark topic-set, avoiding to reward retrieval systems that perform well solely on the most popular topic type.

We analyze several characteristics of the relevant judgements from INEX 2009 for each of the topic classes defined in our classification. The aim of the study is to investigate whether the differences between topic classes could be used to decide on specific retrieval strategies for each of the topic types, a first step towards query-type based focused retrieval.

We also describe the experiments performed on the INEX 2010 data for two different tracks: ad-hoc and data-centric. Our experiments for the ad-hoc track study the performance effects of changing the article order (fetching phase) while in the data-centric track we experiment with the use of different indices and retrievable element types.

The rest of the paper is organized as follows: In Section 2 we present our INEX topic classification and explain the dimensions used. The analysis performed on INEX 2009 data is described in Section 3. Section 4 presents the results of our experiments for both tracks: ad-hoc and data-centric. Finally, we discuss the main contributions and future work in Section 5.

## 2 INEX topic classification

In this section we propose a classification of INEX topics. We extend our previous work on classification of INEX information needs [2] with a new dimension: the type of information sought.

### 2.1 Dimensions

Our classification uses the following three dimensions: 1) the type of information sought (general or restricted), 2) the specificity of the topic (generic or specific), and 3) the complexity of the topic (simple or compound). Topics that are restricted regarding the type of information sought can be further divided according to the type of restriction (topical or structural).

**Type of information sought.** In an XML retrieval scenario, where focused access to relevant information is provided, users can pose restrictions on the type of information being searched for (e.g., I want images or results).

We classify topics into *General* and *Restricted* requests. ***General* requests** are those that ask for *any type of* information about a topic, without restrictions. Note that the topic might be very specific but the type of information the user wants to see is generic (any type of information about it). ***Restricted* requests** are those in which some type of constraint on the type of information being sought is specified. This constraint can be topical (e.g., I want exercises or experiment results) or structural (e.g., I want references or images). In other words, the restrictions can specify which part of the content has to be returned, e.g., "I like to know the *speed capacity* of vehicles" (not any other information on vehicles) or the type of object that it is returned, e.g., "I like too see *images*

of sunflowers" (not any other information/object about sunflowers). Note that *General* requests are those that are typically used in web search and document retrieval, where the task is to return whole documents or web pages.

**Complexity.** In the *complexity* dimension, two categories are used: *Simple* and *Compound*. ***Simple* requests** are those that ask for information about just one topic or aspect of a topic (i.e., mono-faceted requests). While ***Compound* requests** are those that ask for information about several topics or aspects of the same topic (i.e., multifaceted requests) or want information about the relationship between two topics (e.g. technique A in the field of B or information about A for B).

**Specificity.** In the *specificity* dimension, we classify requests into *Specific* and *Generic*, depending on the topical broadness of the information being searched for. In other words, ***General* requests** are those that ask for information about a broad topic, while ***Specific* requests** are those that ask for information about a narrow topic. Note that here we talk about the information being searched for and not about the *type* of information being searched for.

While the *specificity* and the *complexity* of the request are dimensions that have already been used to classify standard IR requests [5], the *type of information sought* is specific to focused retrieval.

Notice also the difference between restricting a topic (e.g., classical movies) and restricting the type of information sought (e.g., pictures of movies). The first one would be *specific* while the second one would be *structurally restricted*.

We hypothesize that the characteristics of the relevant information between the classes of each dimension differ. If these differences exist, XML retrieval systems should use this information in order to optimize their retrieval performance by using specific retrieval strategies for each of the topic classes.

### 2.2 Data classification

The INEX ad-hoc topics are created by the participants following precise instructions. Candidate topics contain a short CO (keyword) query, an optional structured CAS query, a phrase title, a one line description of the search request, and a narrative with details of the topic of request and the task context in which the information need arose. An example of an INEX topic with all its fields can be seen in Table 1. We used the description field of the topics to classify the INEX 2009 topics into different classes. If needed, we used the narrative field from the topic to clarify. All 68 topics were classified by two different volunteers. Table 2 shows the resulting topic classes, the number of topics belonging to each class and gives an example for each of them. We also investigate how *intuitive* the dimensions and categories used in our classification are. We do so by analyzing the level of agreement between volunteers. Table 3 shows the agreement on each of the dimensions between the two volunteers. We can see that the agreements

**Table 1.** Example of INEX topic

| | |
|---|---|
| Topic ID | 2009011 |
| Title | olive oil health benefit |
| CAS query | //food[about(., olive oil) and about(., health benefit)] |
| Phrase title | "olive oil" "health benefit" |
| Description | Find information about what sort of health benefit olive oil has |
| Narrative | I'm a health/beauty buff. I recently learned that olive oil is "good for you". What are the specific health/beauty benefits for consuming olive oil? Any article that mentions health benefits of olive oil is fine, EXCEPT those in which the claim is based on either unpopular/obscure or unscientific methods. So for example, if XXX diet recommends consuming olive oil then it's irrelevant. Note that since Mediterranean diet is not a weight-loss fat diet, but the traditional Mediterranean ways of eating, an article describing the health benefits of olive oil in this setting is relevant. Anything outside of health benefit is irrelevant, how olive oil is produced, the different grades of olive oil etc. |

on the first two dimensions are rather high, suggesting that these dimensions are quite intuitive and objective. However, the specificity dimension has a very low agreement percentage and it is probably too subjective to be used in a real setting.

## 3 Relevance assessments analysis

In this section we investigate whether the characteristics of the relevance judgements differ between the different topic classes described above. Having in mind that for some classes the number of topics is generally too low to draw statistically significant conclusions, we analyze INEX 2009 relevance judgements and look at the relevance characteristics of each of the topic classes. We analyze the following characteristics: 1) the number of relevant documents, 2) the density of the relevant documents, and 3) the number and size of the relevant fragments.

**Number of relevant documents.** By number of relevant documents we refer to the number of unique documents in the collection that contain relevant information given a topic, even if the fraction of relevant information is small. Figure 1 (upper part) shows the average number of documents containing relevant information for each of the topic classes. We can clearly see that *restricted* requests tend to find much less relevant documents than the *general* ones. On average, there are 18 relevant documents for the *restricted* topics and 75 for the *general* ones (13 and 56.5 when looking at the median). Although the difference is not that big for the other dimensions, we see that *compound* requests tend to find less relevant documents than *simple* ones (51 vs. 81 on average and 26 vs. 58 when looking at the median). *Specific* topics are also satisfied with a smaller

**Table 2.** Number of INEX 2009 topics belonging to each of the topic classes and example of topic description for each of them.

| Dimension | Class | Num. | Example |
|---|---|---|---|
| Type of information sought | General | 64 | Information about Nobel prize. |
| | Restricted (structurally) | 2 | Explain "mean average precision" and "reciprocal rank" with **images** or **plots**. Provide **references** in proceedings and journals. |
| | Restricted (topically) | 2 | I want to know vehicles and its **speed capacity** |
| Complexity | Simple | 46 | Information about classical movies |
| | Compound | 22 | Find information about applications of bayesian networks in the field of bioinformatics |
| Specificity | Generic | 13 | I want to find some information about IBM computer |
| | Specific | 55 | Find information on causes of vitiligo and treatment for it |

**Table 3.** Agreement between the two volunteers that classified the topics

| Type of information sought | Complexity | Specificity |
|---|---|---|
| 94% | 85 % | 46% |

number of documents than *generic* ones (65 vs. 100 on average and 48 vs. 65 when looking at the median).

These tendencies are not surprising, it seems reasonable that the more complex, restricted, and specific a topic is, the more difficult is to find information that satisfies it.

**Density.** We also analyze how densely relevant are the documents that contain relevant information. According to recent work [6], focused search works better on sparsely dense documents. We define density of a document as the percentage of relevant text contained in that document (i.e., ratio of relevant text to all text). Text size is given by the number of characters.

Figure 1 (bottom part) shows the average density of the documents containing relevant information for each of the topic classes. We can see that documents that contain relevant information for the *restricted* topics tend to be sparsely dense. On average, 18% of the text in a document is relevant for the *restricted* topics while that is 44% for the general ones (19% and 38% when looking at the median). Focused retrieval seems to be more desirable for *restricted* topics.

Regarding the other dimensions, we see that there is not much difference in terms of density between the *compound* and *simple* topics. In both cases, documents are quite dense on average. The difference is bigger in the specificity dimension. While *generic* topics tend to be answered with highly dense docu-

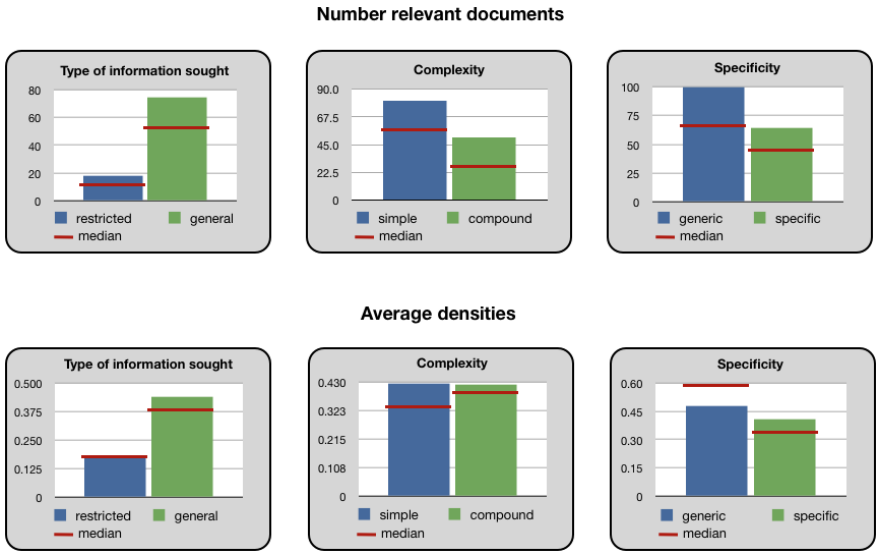



**Fig. 1.** Average number of documents containing relevant information (top) and average density of those documents (bottom)

ments (on average 48% and median 59%), specific topics tend to be answered with less dense documents (41% on average and median 34%).

**Number and size of relevant fragments.** To see how the relevant information is distributed within an article, we look at the number and size of relevant fragments, the fragments that contain the relevant information. Figure 2 shows this information for each of the topic classes. While there are not big differences in the number of relevant fragments between the *restricted* topics (average 2, median 2) and the *general* topics (average 1.7, median 1.5), the fragments for the *restricted* topics tend to be much smaller (see Figure 2 bottom part). On average, relevant fragments for the *restricted* topics are 540 characters long (median 512) while the average length for the *general* topics is 2668 (median 1644).

This is not the case for the other two dimensions where the number and average size of the relevant fragments are very similar between classes. We can see that, in general, a very small number of very long fragments are assessed as relevant, not the best scenario for focused retrieval.

We also look at two characteristics of the topic itself: 1) the number of query terms and 2) the type of CAS query. If there are differences between topic classes, these characteristics can help to automatically classify topics.

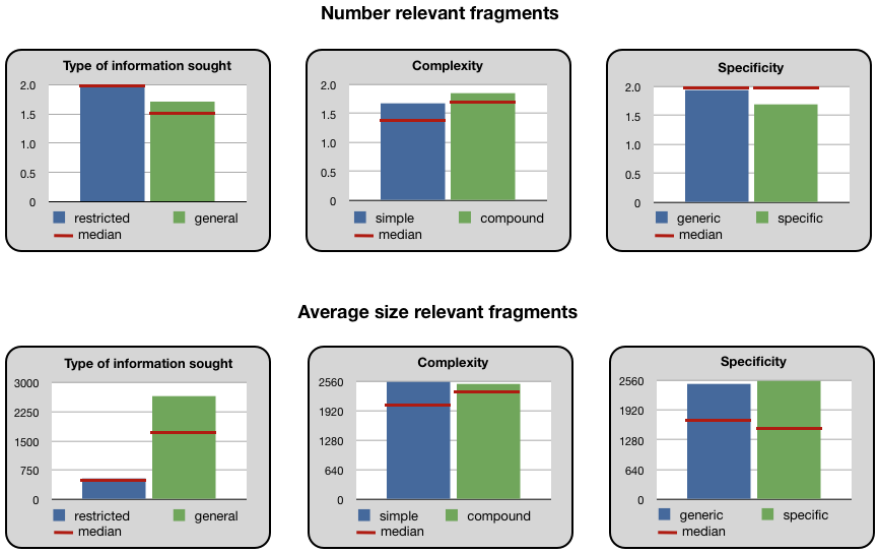



**Fig. 2.** Number and average size of relevant fragments

**Number of query terms.** By number of query terms we refer to the number of terms in the title field of the topic after removing stop-words. Table 3 (upper part) shows the average number of query terms for each of the topic classes. *Restricted* topics tend to be long, they have an average of 6.5 terms per topic (median 6) while the average number of query terms for the *general* ones is 3.7 (median 4). The difference can be explained from the fact that *restricted* requests specify not only what the users are searching for but also the type of information they would like to see. We can see a similar pattern for the specificity dimension. While *generic* topics are expressed, on average, with a very small number of query terms (2.5, median 2), *specific* topics tend to be longer (average 4, median 4). We can also see that there are not big differences regarding the number of query terms between *simple* and *compound* topics (complexity dimension).

**Type of CAS query.** We also analyze which types of CAS query are associated with each of the topic classes. We used five different CAS patterns to classify all topics (see Table 4). Figure 3 (bottom part) shows the percentage of CAS queries of each pattern for each of the topic classes. We can see some differences. While *restricted* topics tend to be specified with longer and more specific CAS queries, the majority of the *general* ones use the most generic form of CAS query[1]. More surprising is that a big portion of the *simple* and *generic* topics use the more

[1] Note that topics that were submitted without a CAS query were assigned the most generic one: //article[about(.,X)].

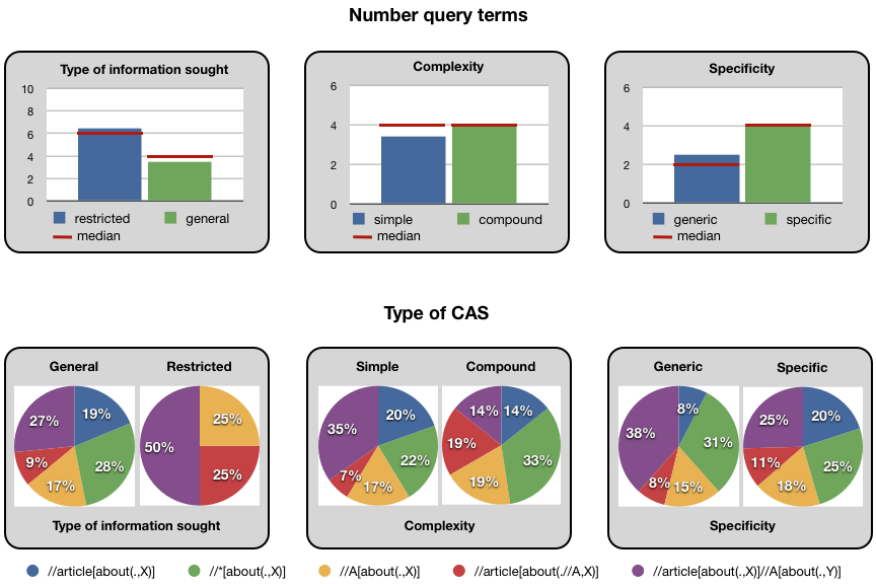



**Fig. 3.** Average number of query terms and percentages of CAS query types

**Table 4.** CAS query patterns

| Pattern | Meaning |
|---|---|
| //article[about(.,X)] | Return articles about topic X |
| //*[about(.,X)] | Return any type of element about topic X |
| //A[about(.,X)] | Return element types A about topic X (where A can be any element type except article) |
| //article[about(.//A,X)] | Return articles that contain an element type A about topic X |
| //article[about(.,X)]//A[about(.,Y)] | Return element types A about topic Y that are contained in articles about topic X. |

specific CAS queries. In general terms however, it is difficult to associate any of the CAS query pattern to a specific topic class.

## 4 Experiments

This section describes the setup and discusses the results of the experiments carried out for the ad-hoc and the data-centric tracks of INEX 2010. For all our experiments we have used the Indri Search Engine [1]. Indri uses a retrieval model based on a combination of language modeling and inference network retrieval frameworks. We have used linear smoothing and varying lambda value. Topics and documents have been pre-processed using the porter stemmer [3] and the smart stop-word list [4].

### 4.1 Ad-hoc track experiments

For the ad-hoc track experiments we have used the Indri search engine [1] with linear smoothing and lambda 0.45. The lambda value has been set to 0.45 after training on the INEX Wikipedia 2009 collection. The only indexed fields are articles, sections, and paragraphs, meaning that only these element types can be explicitly retrieved. We study the importance of the fetching phase, i.e., the performance effects of changing the article order.

**Relevant in Context** The aim of the Relevant in Context Task is to first identify relevant articles (the fetching phase), and then to identify the relevant results within the fetched articles (the browsing phase). As mentioned above, we experiment with the performance effects of the fetching phase. For that, we use the same baseline run and reorder its articles in three different ways. Our baseline is a paragraph run (retrieving only paragraphs) grouped by article. The final article order is given by 1) our own article run order (retrieving only articles), 2) the reference run order, 3) the baseline run order (i.e., the article where the most relevant paragraph appears, followed by the article were the second most relevant paragraph appears, etc.).

Our original submissions were not valid due to a bug in our code and could not be evaluated. The results of the *un-official* runs (after fixing) are shown in Table 5. We can see that changing the article order affects considerably the

**Table 5.** Un-official results for the relevant in context runs. The number in parentheses indicates the estimated run position in the official ranking.

| run name | MAgP (est. position) |
|---|---|
| UPFpLM45coRC1 | 0.1109 (34) |
| UPFpLM45coRC2 | **0.1571 (9)** |
| UPFpLM45coRC3 | 0.0763 (39) |

performance of the run. The reference run article order outperforms the other two. Paragraphs are not good indicators of article relevance. Ranking articles by their most relevant paragraph performs the worst.

**Restricted Relevant in Context** The Restricted Relevant in Context Task is a variant of the Relevant in Context task, where only 500 characters per article are allowed to be retrieved. Overlapping results are not permitted. For this task, we have followed a similar approach to the one of our Relevant in Context runs. This time however, our baseline is a section run (retrieving only sections) and a second post-processing step is made in order to return only 500 characters per article. That is, per article we return the most relevant sections until the 500th character is reached. As in the previous task, the final article order is given by 1) our own article run order (retrieving only articles), 2) the reference run order, 3) the baseline run order (i.e., the article where the most relevant section appears, followed by the article were the second most relevant section appears, etc.).

As in the previous task, our original submissions were not valid due to a bug in our code and could not be evaluated. The results of the *un-official* runs (after fixing) are shown in Table 6.

**Table 6.** Unofficial results for the restricted relevant in context runs. The number in parentheses indicates the estimated run position in the official ranking.

| run name | MAgP (est. position) |
|---|---|
| UPFpLM45coRRC1 | 0.0894 (15) |
| UPFpLM45coRRC2 | **0.1210 (13)** |
| UPFpLM45coRRC3 | 0.0633 (17) |

We see similar performances as in the previous task; the reference run article order outperforms the other two. In absolute numbers paragraph runs seem to outperform section runs. However, the lower performance of the section runs could be due to the task restrictions. When looking at the relative position of the runs regarding the other groups, the three runs performed relatively well (top 15-20).

**Restricted Focused** The Restricted Focused task aims at giving a quick overview of the relevant information in the whole of Wikipedia. Results are restricted to max. 1,000 characters per topic. For this task, we return a single paragraph per article (the most relevant) until we reach the 1,000 characters per topic. The assumption is that users prefer to see an overview based on the largest number of articles rather than seeing several relevant paragraphs of the same article. Our three official runs are again based on different article order as in the previous tasks; The final article order is given by 1) our own article run order (retrieving only articles), 2) the reference run order, 3) the baseline run

order (i.e., the article where the most relevant paragraph appears, followed by the article were the second most relevant paragraph appears, etc.).

The results of these runs are shown in Table 7. As in the other tasks, we

**Table 7.** Official results for the restricted focused runs. The number in parentheses indicates the run position in the official ranking.

| run name | char prec (position) |
|---|---|
| UPFpLM45coRF1 | 0.2984 (19) |
| UPFpLM45coRF2 | **0.3066 (15)** |
| UPFpLM45coRF3 | 0.1156 (30) |

can see that the article order is an important factor on the overall result of the run. There is a big difference in terms of performance from our article order and the reference run order and our paragraph run order. Paragraphs are not good estimators of the total relevance of an article. In other words, a relevant paragraph does not imply that the article is relevant to the same degree.

### 4.2 Data-centric track experiments

For our data-centric track experiments we used the Indri search engine [1] with linear smoothing and two different lambdas, 0.45 and 0.15. Since this is a new collection and we did not have training data to optimize lambda, we experimented with two different values that have been successfully used in other collections. We also experimented with the use of two different indices (indexing all the collection vs. indexing only movies) and by restricting the type of elements to be retrieved (no restriction vs. movie elements)[2].

Table 8 shows the parameters used for each of our official runs and Table 9 the official results. Our best performing runs are the ones that use the movie index,

**Table 8.** Official runs for the data-centric track

| run name | index | retrievable elements | lambda |
|---|---|---|---|
| UPFL15Tall | all | no restriction | 0.15 |
| UPFL45Tall | all | no restriction | 0.45 |
| UPFL15Tmovie | all | movie | 0.15 |
| UPFL45Tmovie | all | movie | 0.45 |
| UPFL15TMI | movies | no restriction | 0.15 |
| UPFL15TMImov | movies | movie | 0.15 |

[2] Note that movie elements can have very different forms: from a complete movie document to a movie element within a list of movies played by an actor.

**Table 9.** Official results for the data-centric track. The number in parentheses indicates the run position in the official ranking.

| run name | MAgP | MAiP | Document Retrieval |
|---|---|---|---|
| UPFL15Tall | - | 0.1486 (7) | 0.2961 (6) |
| UPFL45Tall | - | 0.1338 (11) | 0.2822 (8) |
| UPFL15Tmovie | - | 0.0770 (20) | 0.1983 (16) |
| UPFL45Tmovie | - | 0.0410 (24) | 0.1578 (20) |
| UPFL15TMI | **0.2459 (2)** | **0.1809 (2)** | **0.3687 (3)** |
| UPFL15TMImov | 0.2434 (3) | 0.1762 (3) | 0.3542 (4) |

indicating that for this specific topic set the use of other types of documents introduces noise. We also see that lambda 0.15 always performs better than lambda 0.45, indicating that it is better to give less emphasis to the collection statistics. Figure 4 show the official graphs. In general terms we can see that using the movie index (our best runs) leads to high precision at early recall levels while, not surprisingly, it does not manage to do so at middle and/or high recall levels (MAiP and MAP graphs). This is because a large part of the collection is not indexed, which makes it difficult, if not impossible, to have a high overall recall.

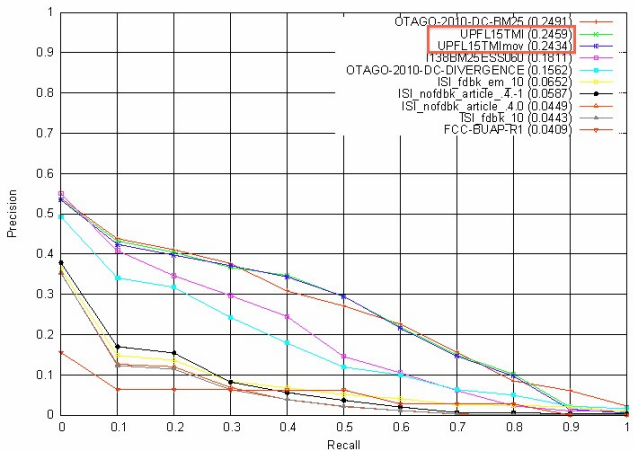
## 5 Discussion and Conclusions

This paper described our participation at INEX 2010. We presented a classification of INEX topics and an analysis of the characteristics of the relevance assessments for each of the topic classes. The goal of our study was to investigate whether there are differences in relevance judgements between topic classes in order to use them for retrieval. We have seen, for instance, that *restricted* topics have a small set of relevant documents which are sparsely dense and relevant information is contained in small fragments of documents. Although some of the analyzed relevance characteristics differ between classes, it is not clear whether this information could be used for retrieval. More data needs to be analyzed in order to see whether these differences are statistically significant.

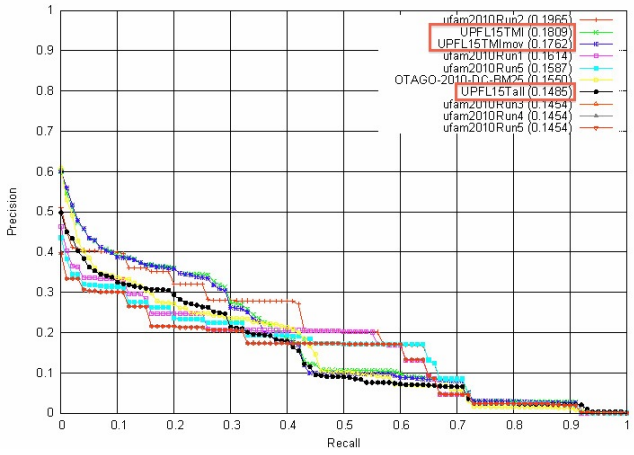
The classification presented is based on three different dimensions (type of information sought, complexity, and specificity), generic enough to be used in other focused retrieval scenarios. Our goal is to use it to perform different retrieval strategies for each of the topic classes. A classification can also help to provide a more balanced benchmark topic-set, avoiding to reward retrieval systems that perform well solely on the most popular topic type. We note that not all the dimensions are objective enough to be easily used. The specificity of a topic is a subjective matter and it might not be easy to apply in real settings.

As future work we plan to investigate whether it is beneficial to use different retrieval strategies for the different topic types.

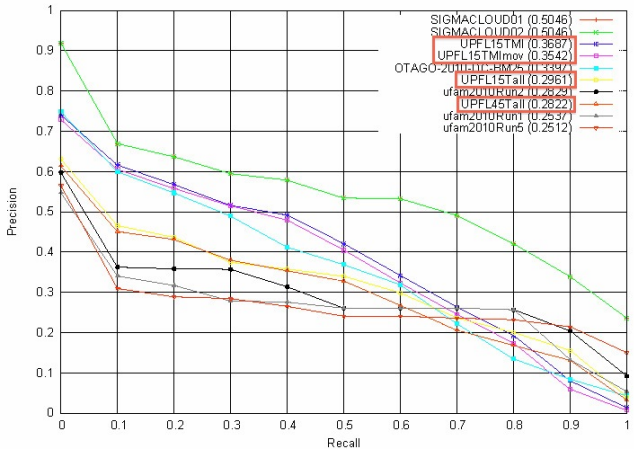
We reported on our experiments for INEX 2010, in the ad-hoc and data-centric tracks. In the ad-hoc track we studied the performance effects of chang-




MAgP




MAiP




MAP

**Fig. 4.** Official evaluation graphs for the data-centric track

ing the article order (fetching phase) while in the data-centric track, we experimented with the use of different indices and retrievable element types. Our results on the ad-hoc track confirm that article order is a very important factor on the overall performance of the systems. In all of our experiments, the article order of the reference run outperforms the other runs. In the data-centric track, our main finding is that indexing only the movie documents leads to much better performance than indexing the complete collection.

**Acknowledgments** This work has been supported by the Spanish Ministry of Science and Education under the HIPERGRAPH project and the Juan de la Cierva Program.

## References

1. T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, Indri: a language model based search engine for complex queries. Proceedings of the International Conference on Intelligent Analysis, 2005.
2. G. Ramírez Camps, Structural Features in XML Retrieval. PhD thesis, University of Amsterdam, 2007.
3. M.F. Porter, An algorithm for suffix stripping. Program, 14(3) pp 130?137, 1980.
4. G. Salton, The SMART Retrieval SystemExperiments in Automatic Document Processing. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1971.
5. P. Ingwersen and K. Järvelin, The Turn: Integration of Information Seeking and Retrieval in Context. Springer, The Information Retrieval Series, volume 18, 2005.
6. P. Arvola and J. Kekalainen and M. Junkkari, Focused Access to Sparsely and Densely Relevant Documents Proceedings of SIGIR 2010, 2010.