# From natural language to NEXI, an interface for INEX 2005 queries

Xavier Tannier

École Nationale Supérieure des Mines de Saint-Etienne
158 Cours Fauriel
F-42023 Saint-Etienne, France
tannier@emse.fr

**Abstract.** Offering the possibility to query any XML retrieval system in natural language would be very helpful to a lot of users. In 2005, INEX proposed a framework to partipants that wanted to implement a natural language interface for the retrieval of XML documents, independantly of the search engine. This paper describes our contribution to this project and presents some opinions concerning the task.

## 1 Introduction

### 1.1 Motivation

Asking a question in everyday language ("natural language") and getting a relevant answer is what the everyday user really miss in the process of Information Retrieval (IR). Moreover, as natural language is the best way so far to explain our information need, using it should help a system if the query was analysed correctly. However, at present, Natural Language Processing (NLP) techniques are not developed enough to come close to the human perception of language, and actual results are not yet up to what we could expect [1, 2].

In the case of "traditional" IR, where documents are considered as text only (*flat* documents), classical search engines need a query composed of a list of keywords. Writing such a query is quite simple for the casual user, and the value added by NLP approaches is not worth the complexity of these techniques.

On the other hand, many natural language interfaces (NLI) for querying structured documents (databases) have been developed, most of them transforming natural language into Structured Query Language (SQL) [3, 4, 5]. This is probably because the benefits that can be gained in that case are much higher than in traditional information retrieval. Indeed, SQL (and any structured query language used for XML retrieval as well) is hardly usable by novice and casual users. Moreover such languages impose to know the structure of the database (or of the documents).

But database querying is a strict interrogation. It is not information retrieval. The user knows what kind of data is contained in the database, the information need is precise, and a correct query necessarily leads to a correct answer. This means that the natural language analysis must interpret the query perfectly and unambiguously, failing which the final answer is incorrect and the user disatisfied. For this reason notably, natural language interfaces for databases only apply to very restricted domains. Even in these domains, the answer to a query is often *"I did not understand your query"*.

XML retrieval stands between these two domains (see Tab. 1). *Document-oriented* XML files [6], as well as databases, contain some structural information, and the use of a NLI would be justified. But in XML IR, as in traditional IR, the information need is loosely defined and there is no perfect answer to a query. A NLI is then a part of the retrieval process, and thus it can interpret some queries imperfectly, and still return useful results. The problem is then made "easier" to solve... and we can even imagine an interface getting better results than manual queries (which is a nonsense in databases). Moreover more general applications can be designed. In return, such an interface has to be very robust, and all queries must be analysed, even imperfectly. It is not conceivable that the system returns no answer because it did not understand the question.

**Table 1.** Some features of flat, semi-structured and structured documents in an Information Retrieval point of view.

| | *Flat documents* | *semi-structured documents (XML)* | *structured documents (DB)* |
|---|---|---|---|
| *Content* | text only | text + structure | structure + data |
| *Information need* | text only | content and/or structure | |
| *Query* | keywords | **Structured query languages** | |
| *Interpretation* | **loose (IR)** | | strict |

### 1.2 INEX and Natural Language Tasks

The INitiative for Evaluation of XML Retrieval (INEX) aims at evaluating the effectiveness of Information Retrieval systems for XML documents. The INEX collection groups a set of 16819 articles from the IEEE Computer Society, represented in XML, with a set of topics and human assessments on these topics.

In 2005 campaign, two different types of topics have been designed [7]:

- *Content Only + Structure* (CO+S) topics, as indicated by their name, refer only on textual content, but the user can nevertheless add some structural hints to help the system.
- *Content And Structure* (CAS) topics allow a user that know the structure of the documents to formulate constraints on structural elements that he/she wants to be searched for.

We participated to the campaign for both categories, but our approach focuses principally on CAS topics. A simplified example of INEX 2005 topic is given in Fig. 1. The element `castitle` is written in NEXI [8], a formal language for XML retrieval.

```
<inex_topic topic_id="203" query_type="CO+S" ct_no="5">
    <title>code signing verification</title>
    <castitle>//article//sec[about(., code signing verification)]</castitle>
    <description>
        Find documents or document components, most probably sections, that
        describe the approach of code signing and verification.
    </description>
    <narrative>
        I am working in a company that authenticates a wide range of web data
        base applications from different software vendors. [...] To be relevant,
        a document or document component must describe the whole process of
        code signing and verification, which means [...]
    </narrative>
</inex_topic>
```

**Fig. 1.** Example of INEX 2005 topic. The `title` element is used for Content-Only search, `castitle` for structural hints and CAS representation in NEXI. `description` is used by Natural Language Processing tasks participants, while the `narrative` is reserved for human assessors.

NEXI CAS queries have the form **//A[B]//C[D]** where A and C are paths and B and D are filters. We can read this query as *"Return C descendants of A where A is about B and C is about D"*. B and D correspond to disjunctions or conjunctions of 'about' clauses **about(//E, F)**, where E is a path and F a list of terms. The `'title'` part of Fig. 1 gives a good example of a query formulated in NEXI. More information about NEXI can be found in [8].

In 2005 INEX campaign, two different tasks aimed to involve Natural Language Processing. In the first one, called NLQ (Natural Language Queries), participants had to consider only the `description` part of the topics and to return a set of XML elements (or doxels) corresponding to the request. No matter how they performed their search, or where the NLP was used. The evaluation of NLQ systems was the same as for the *ad-hoc* task.

In the second one, NLQ2NEXI, on which this paper focuses, the aim was to translate natural language queries into `title` (keyword list) and `castitle` (NEXI) elements from the `description`. Here the idea is to build a generic interface that could used by any retrieval system reading NEXI queries. Automatically generated topics have then been run with a search engine $\mathcal{S}$ provided by the organizers (Fig. 2). In this case, the evaluation is twofold:

1. a comparison between the effectiveness of each NLQ2NEXI system.
2. a comparison between each system and a baseline obtained by running the system $\mathcal{S}$ on initial (manual) topics, in order to quantify the trade-off in performance.

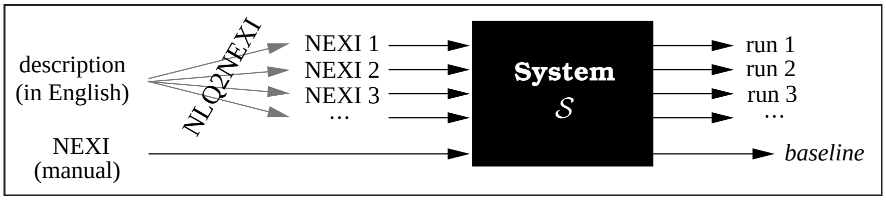

**Fig. 2.** NLQ2NEXI.

## 2 Natural Language Query Analysis

In our approach, requests are analysed through several steps:

1. A part-of-speech (POS) tagging is performed on the query. Each word is labeled by its word class (*e.g.:* noun, verb, adjective...). To carry out this task we chose the open-source free tool TreeTagger [9].
2. A POS-dependant semantic representation is attributed to each word. For example the noun *'information'* will be represented by the predicate *information(x)*, or the verb *'identify'* by *evt(*$e_1$*, identify).*
3. Context-free syntactic rules describe the most current grammatical constructions in queries and questions. Low-level semantic actions are combined with each syntactic rule. Two examples of such operations, applied to the description of Topic 130 (INEX 2004: *"We are searching paragraphs dealing with version management in articles containing a paragraph about object databases."*), are given in Fig. 3. The final result is a logical representation shown in the left part of Fig. 4. This representation is totally independant from the queried corpus, it is obtained by general linguistic operations.
4. The semantic representation is then reduced with the help of specific rules:
   - a recognition of some typical constructions of a query (*e.g.: Retrieve + object*) or of the corpus (*e.g.: "an article written by [...]"* refers to the tag *au − author*);
   - and a distinction between semantic elements mapping on the structure and, respectively, mapping on the content;

   Figure 4 shows the specific rules that apply to the example.
5. A treatment of relations existing between different elements;
6. The construction of a well-formed NEXI query.

Steps 1 to 5 are explained in more details in [10], as well as necessary corpus knowledge and the effect of topic complexity on the analysis. The representation obtained at the end of Step 5 does not depend on any retrieval system or query language. It could be transformed (with more or less information loss) into any existing formal language.

Transformation process from our representation to NEXI is not straightforward. Remember that a NEXI query has the form **//A[B]//C[D]**.

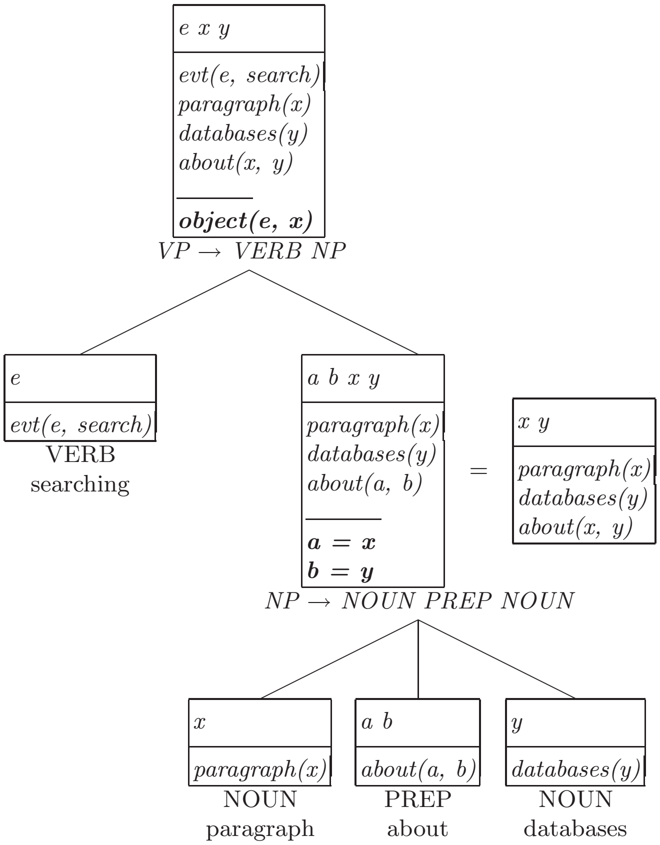

**Fig. 3.** Example of rule application for the verbal phrase "*searching paragraphs about databases*" (rules *NP* → *NOUN PREP NOUN* and *VP* → *VERB NP*). Basic semantic representations are attributed to part-of-speeches (leaf components). When applying syntactic rules, components are merged and semantic actions are added (here identity relations and verbal relation predicate – bold predicates).

- At content level, linguistic features (like `noun_modifier` in the example) cannot be kept and must be transformed in an appropriate manner (see Sect. 3).
- At structural level, a set of several tag identifiers (that can be DTD tag names or wildcards) has to be distributed into parts A, B, C and D, that we respectively call support requests, support elements, return requests and return elements. These four parts A, B, C and D are built from our representation (Fig. 4) in the following way:
  - C is the 'framed' (selected) element name (see Fig. 4 and its caption);
  - D is composed of all C children (relation *contains*) and their textual content (relation *about*);

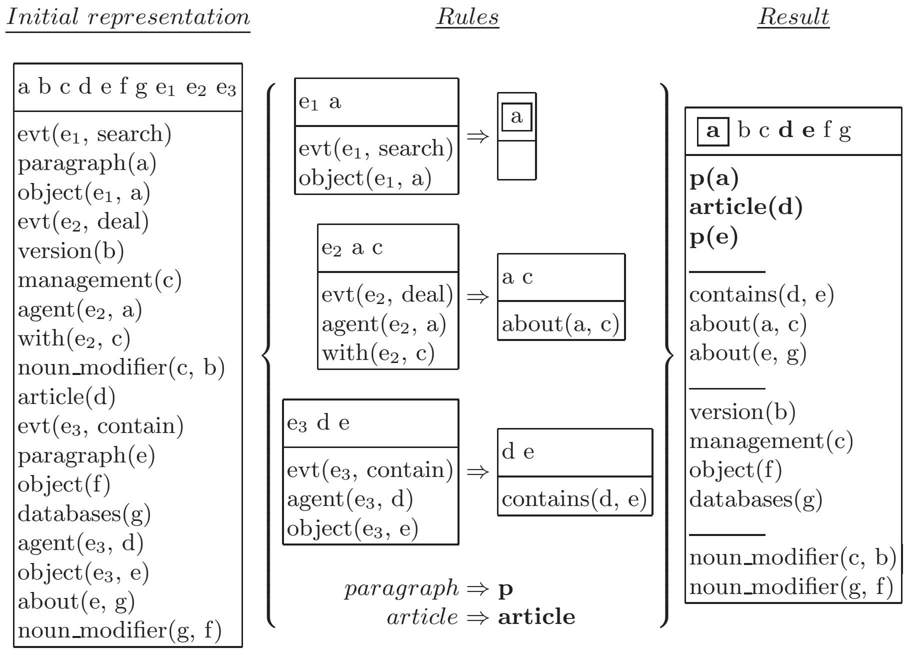

**Fig. 4.** The semantic analysis of Topic 130 (left), is reduced by some generic rules (center), leading to a new representation (right). Bold predicates emphasize words representing XML tag names and the framed letter stands for the element that should be returned to the user. The first three rules deal with verbal phrases *"to search sth"*, *"to deal with sth"* and *"to contain sth"*.

- A is the highest element name in the DTD tree, that is not C or one of its children;
- B is composed of all other elements and their textual content.

Wildcard-identified tags of the same part are merged and are considered to be the same element. See an example in Sect. 4.

## 3 Noun phrases

Our system generates some linguistic-oriented predicates. The main ones are `np_property`, `noun_modifier` and `adjective`. NEXI format requires the 'about' clauses to contain only textual content. Phrases can be represented with quotation marks. We chose to consider only noun phrases treatment here, because other relations are translated in a straightforward way.

From an IR point of view, noun phrases have the general form [11]:

$$NP \rightarrow det^{*}\ pre^{*}\ head\ post^{*}$$

...Where *det* is a determiner, *pre* (*premodifier*) is an adjective, a noun or a coordinated phrase, *head* is a noun and *post* (*postmodifier*) is a prepositional phrase or a relative clause.

In our representation, relations between premodifiers and head nouns are expressed by predicates `noun_modifier` (if the premodifier is a noun) or `adjective` (if the premodifier is an adjective). Prepositional relations between NPs (*i.e.* the form $NP \rightarrow NP_{head}\ PREP\ NP_{post}$) are represented by `noun_property`.

All forms have been considered when analysing the natural language queries, but we distinguished two specific constructions of NPs to build the formal queries.

### 3.1 Simple noun phrases

In English, the simplest noun phrases are a succession of adjectives or nouns followed by a head noun:

$$NP \rightarrow (ADJ — NOUN)+\ NOUN \qquad (1)$$

These multi-word terms are less ambiguous than simple nouns, and generally refer to a particular domain [12]. They are not subject to many syntactical variations (see next section), and it is quite probable that such terms representing the same concept have the same form in most occurrences of a collection. For all these reasons, these simple NPs are very interesting in information retrieval. Some examples (extracted from INEX 2005 topics) are given in Tab. 2 (with an additional rule for proper names: $NP \rightarrow PN+$).

With NEXI, phrases are represented between quotation marks. All sequences of words obeying to Rule 1 are then transcribed between quotation marks.

**Table 2.** Examples of simple noun phrases (rule 1) in INEX 2005 topics.

| Topics | Noun phrase |
|---|---|
| *204* | “semantic networks” (ADJ NOUN) |
| *231* | “graph theory” (NOUN NOUN) |
| *210* | “multimedia document models” (NOUN NOUN NOUN) |
| *211* | “global positioning systems” (ADJ NOUN NOUN) |
| *204* | “Dan Moldovan” (PN PN) |

### 3.2 Complex noun phrases

Nouns or noun phrases linked to each other by prepositions are semantically very significant [13, 14]:

$$NP \rightarrow NP\ (PREP\ NP)+ \qquad (2)$$

They occur as frequently as constructions made with Rule 1 (see Tab. 3). However it is quite hazardeous to consider them as a unique multi-word term in the same way. In particular, they are subject to many variations in their form. *Fabre and Jacquemin* [15] distinguished five different simple syntactic forms that could represent the same concept in French. For example, even without semantic variation (as synonymy), the NP *"annotation in image retrieval"* found in Topic 220 can be modified with no or little semantic change into *"annotate images for retrieval"*, *"retrieve annotated images"*, *"annotated image retrieval"*, *"retrieval of annotated images"*, *"images have been annoted for retrieval"*, etc.

**Table 3.** Examples of complex noun phrases (Rule 2) in INEX 2005 topics.

| Topics | Noun phrase |
|---|---|
| *208* | history of Artificial Intelligence |
| *216* | the architecture of a multimedia retrieval system |
| *217* | user-centered design for web sites |
| *219* | the granularity of learning objects |
| *220* | annotations in image retrieval |
| *233* | development of synthesizers for music creation |
| *276* | evaluation measure for clustering |

Moreover such a phrase does often not occur at all in a relevant element. In a phrase having the form *"$NP_1$ PREP $NP_2$"*, we have noted that one of the sub-NPs represents the *context*, while the other one represents the *subject* of the current sentence. The role of each part depends on the structure of the document.

For example, suppose we look for an element dealing with *"evaluation measure for clustering"* (Topic 276). In an article about *clustering* on the whole, we just need to look for the term *"evaluation measure"*. Inversely, an article about *evaluation measures* in general must contain an element treating *"clustering"*.

We have noted, after 2004 campaign, that this issue was an important source of mis-retrieval for search engines. In the case of topic descriptions containing $NP_1$ *PREP* $NP_2$, where $NP_2$ was the context in most documents, many retrieved doxels contained $NP_1$ in a bad context, and then were not relevant. For example, a search for *"navigation systems for automobiles"* (Topic 128) returned many doxels about navigation systems in planes or ships in the first ranks.

In this case, to remedy this problem, we would like to perform a *contextual research* (for *"navigation systems"* in the context of a section or an article about *"automobiles"*, or inversely), but also a *conditional research* within a single doxel (if a doxel is relevant with *"automobiles"*, then check for *"navigation systems"*).

Unfortunately this kind of features can hardly be represented with a single NEXI query. Even so we tried to simulate such a behaviour. We noticed that the most frequent configuration was "$NP_1$ in the context of $NP_2$" when the topic description contained a $NP_1$ *PREP* $NP_2$ phrase. We decided to translate such NPs in the following way:

– Contextual search: Addition of $NP_2$ into a support part concerning the whole article (root element).
– Conditional search: Addition of a sign '+' before $NP_2$ in the current part.

For example, *"a paragraph about navigation systems for automobiles"* can be translated into:

```
/article[about(., automobiles)]//p[about(., ‘‘navigation systems’’)
                                   AND about(., +automobiles)]
```

In our tests with INEX 2004 collection, this approach led to a increase in precision of about 10 %. But then the support element construction is quite artificial, and this is done to the detriment of strict evaluation metrics (strict quantization and strict interpretation of target and/or support element requirements [16]). By choosing this strategy we admit that we focus principally on vague interpretation and generalised quantization.

## 4 Example

We give here a significant example, with the analysis of a slightly simplified version of Topic 219 (INEX 2005). Several syntactic parsings could be possible for the same sentence. In practice a "score" is attributed to each rule release, depending on several parameters. In our sample topic only the best scored result is given.

(219) Find sections that discuss the granularity of learning objects.

Figure 5 shows the three major steps of the analysis of this topic. The left frame represents the result of Step 3 (see Sect. 2). Some IR- and corpus-specific reduction rules are then applied and lead to right frame: the term *section* is recognized as tag name **sec** (line 3); the construction *"c2 discusses c4"* is changed into **about(c2, c4)** (lines 4 to 6). The other relations are kept. Translation into NEXI is performed as explained above.

## 5 Results

We present here our results for both CO+S and CAS INEX 2005 tasks. For CAS task, different evaluations have been performed, depending on the interpretation of structural constraints (vague or strict [16]). Different sets of metrics were also used. In Fig. 6, we chose four curves illustrating NLQ2NEXI evaluation with the nxCG metric and generalised quantization [16]. These samples, representative of the entire result set[1], show the comparison between the baseline and our own best run. We also stress the general best run (when not ours).

[1] Excepted for FetchBrowse CO sub-task, where all participants failed to approach the baseline.

c1 c2 c3 c4 c5 c6

1. event(c1, find)
2. object(c1, c2)
3. section(c2)
4. event(c3, discuss)
5. agent(c3, c2)
6. object(c3, c4)
7. granularity(c4)
8. rel_np_relation(c4, c6, of)
9. object(c6)
10. learning(c5)
11. rel_noun_modifier(c6, c5)

$\xrightarrow[rules]{reduction}$

$\boxed{c2}$ c4 c5 c6

**sec(c2)**

about(c2, c4)

granularity(c4)
object(c6)
learning(c5)

rel_np_relation(c4, c6, of)
rel_noun_modifier(c6, c5)

```
//article[about(., "learning objects")]//sec[about(., granularity) AND
about(., +"learning objects")]
```

**Fig. 5.** Semantic representations of Topic 219, and automatic conversion into NEXI.

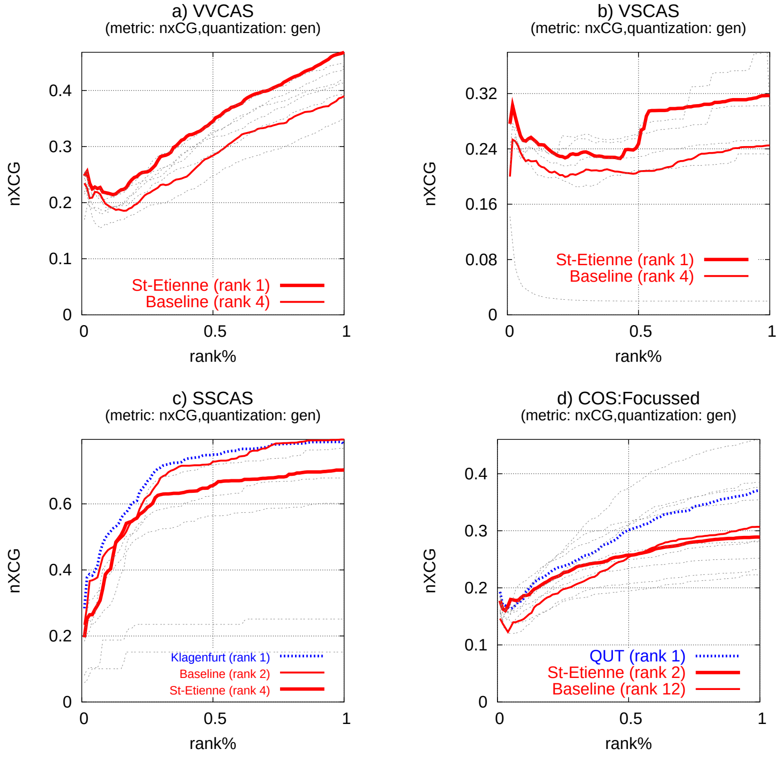

**Fig. 6.** Official NLQ2NEXI results for VVCAS, VSCAS, SSCAS and COS.Focussed tasks, normalized extended cumulated gain at 50, generalised quantization [16].

These results show a very good performance of our system for CAS task, especially for vague interpretation (as anticipated in Sect 3.2). Baseline is outperformed in most of the case, and often widely. This is a strong improvement in comparison with previous year participants results, where the baseline performed about 20% better than natural language systems.

CO+S results are also good, since baseline is still below our system, even if the difference is not as large. Anyway it is now proved that a automatic translation can do better than a manual process.

## 6 Limits

### 6.1 Limits of the task

Translation of natural language queries into a formal language like NEXI encounters some limits, mainly due to the fact that the natural language interface cannot give some specific instructions to the retrieval system. The formal language, if not especially designed for this aim, is a pivot preventing from any “communication” between both systems. For example, it is not possible to consider the following features within single NEXI queries[2]:

- NEXI does not allow to perform any conditional search (see Sect. 3.2). The use of '+' sign is not semantically reliable and is often not considered by search engines.
- NEXI cannot either deal with contextual search: A reference to the context occurs preferentially before the retrieved element, in the paragraph preceding it, or in the introduction of the section, etc. Directly refering to the article as a support part of the query (as we did) is too vague.
- NEXI does not bring any proximity operators for terms or structure (but retrieval models can compensate, many systems allow a flexible treatment of phrases [17], and some consider the proximity of doxels [18]).
- It is not possible to represent non-hierarchical relations between elements with NEXI (precedence for example).
- Finally, NEXI is only a query language. It is not designed to deal with any linguistic features. With the linguistic analysis, the interface finds some interesting relations between terms (or elements), as semantic relations (agent, object, etc.), but the translation forces us to give this knowledge up.

On the one hand, formal languages will always stay more precise than natural languages. If sometimes the system with a NLI outperforms the same system with a hand-made NEXI query, this is because the interface found a better, more complete and/or more adequate way to represent the information need. On the other hand, for all the reasons above, the use of these formal languages, if they are not thought with this aim in mind, leads to some loss of information.

[2] In addition to this list, NEXI is not designed to deal with many database-oriented constraints, but we are not either interested by this aspect.

If an interface is very interesting because it can be "plugged" to (hopefully) any kind of formal languages, and then be applied to many existing systems, it would also be worthwhile to go further and to build a system with a self-made pivot or without pivot at all.

### 6.2 Limits of the evaluation

In the *ad-hoc* task, the input is constant and the retrieval systems are different. To evaluate these systems we look at their output (a ranked list of XML elements). In NLQ2NEXI task, the challenge is precisely to produce the input, and the evaluation is performed indirectly, through the use of a search engine that is common to all participants (different inputs, same system). This way we can make sure that the differences in retrieval performance are really due to the quality of the input, and it becomes possible to compare all NLQ2NEXI systems with each other.

Another way to evaluate interfaces is to compare them with a manual baseline. The same system is run on official NEXI queries[3] (manually written by the author of each topic). In this case, automatic processes are compared with a manual process. Like all human interventions (and IR evaluation is full of them), this introduces a new bias: automatic systems are compared with a query built by a given person at a given time. Probably some different manual translations of the topic description would have led to better results. Moreover, many CO+S topics do not have any NEXI `castitle`[4].

Finally, manual translations from description to NEXI are not always faithful, even if this is much better than it was in 2004 [19]. In particular, many CAS subtopics seem to have a problem with NEXI constraints. Topic 251 is characteristic of this issue:

(251) We are searching paragraphs which are descendant of a section dealing with web information retrieval.

In the official transcription of the description into NEXI, the paragraphs are considered to be dealing with web information retrieval:

```
//article//sec//p[about(., web retrieval)]
```
[5]

Even if this interpretation is syntactically correct, it seems obvious that any human people would understand that the section is concerned by the verbal phrase (*dealing with web information retrieval*):

[3] What we call the "official" NEXI title of a topic is the query proposed by the author (see the example of Fig. 1). This NEXI query is used by the INEX *ad-hoc* task participants.

[4] Besides, a study on the performances of automatic NEXI titles compared to CO official titles would probably be very interesting.

[5] By the way we can note that *"web information retrieval"* has been replaced by *"web retrieval"*.

```
//article//sec[about(., web retrieval)]//p
```

But this form is not correct in NEXI (where the returned element must contain an *about* clause [8]).

The narrative part of this topic confirms the author's NEXI title, but adds to the confusion: *"the paragraphs which are descendants of section describing the topic related to web information retrieval are also regarded as relevant. However, compared with paragraphs described above, these are considered less relevant"*.

### 6.3 Limits of our system

The following is a non-exhaustive list of problems encountered by our natural language interface. In our opinion, these issues represent the most important factors that make the system not work well for some topics. We do not broach here the "usual" difficulties that NLP has in traditional information retrieval (spelling mistakes, noise produced by non query terms, anaphoras, pragmatic issues. . . ) , but rather those that are specific to structural constraints.

**Lexical ambiguity.** Classical lexical problems in IR are semantic relations between different words (synonymy, hyponymy, etc.) and words that have multiple meanings (homographs). Homographs raise a new problem in XML retrieval, where some words can be understood as normal content-based query terms, but also as tag names (or a synonym of a tag name). Using a simple dictionnary of synonyms to detect references to tag names is obviously not enough. For example the words *"document"* and *"information"* are, most of the time, used for refering to XML elements (*"Find information/documents about"*). But how to deal with a query about *"multimedia document models"* (Topic 210) or *"incomplete information"* (Topic 224)? What if the query is *"Retrieve information about information retrieval"*? Actually it does not seem so difficult to handle this with specific syntactic features, but we clearly under-estimated this issue so far.

**Corpus-dependant knowledge.** Throughout the query analysis, we use several kinds of information about the corpus, among which the DTD (and the terms associated with tag names), but also some specific linguistic constructions. For example, as shown by Fig. 7, a query about *"information by Moldovan"* (Topic 204) implicitely refers to an author (tag `'au'` in INEX collection); *"works citing Beaza-Yates"* (Topic 280) introduces a bibliographic element. All these rules are necessary to analyse many queries properly, but are an obstacle to the extension of the tool to general corpora, or to heteregenous collections[6].

[6] Note that these rules are structure-specific, but not domain-specific (in the case of INEX, this means that no rules have been set up especially for computer science information retrieval).

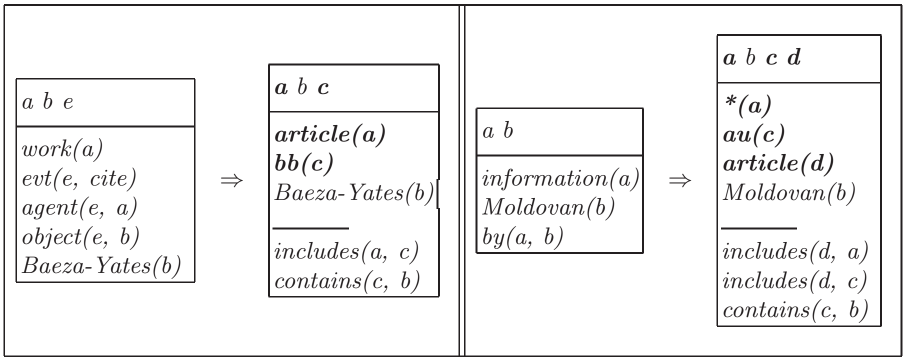

**Fig. 7.** Examples of corpus-dependant rules, applied on *"works citing Baeza-Yates"* (left, Topic 280) and *"information by Moldovan"* (right, Topic 204).

## 7 Conclusion

INEX 2005 NLQ2NEXI task proves that the help brought by an natural language interface is very effective. NEXI queries that are automatically obtained from a description in English lead to better results than manual queries, yet written by experts. This is the proof that natural language explanations of an information need are not only easier to formulate, but also more effective. The results also confirm the assumptions made in the introduction: building a natural language interface for XML retrieval is much different than doing it for database querying or traditional IR.

Moreover, techniques used by participants are quite different; While we (Ecole des Mines de Saint-Etienne) obtain the best scores in CAS with a vague interpretation of elements (6.a and 6.b), University of Klagenfurt performs better in strict interpretation (6.c) and Queensland University of Technology gets its best results in CO task (6.d). Teams have a lot to learn from each other, and global results should improve a lot in the future. But each technique produces good scores for a given task to the detriment of another one, and the best way to progress is probably to define a new model taking all what we need into account (like conditional and contextual searches proposed in this article).

## References

[1] Strzalkowski, T., Lin, F., Wang, J., Perz-Carballo, J.: Evaluating Natural Language Processing Techniques in Information Retrieval. [20] 113–145

[2] Sparck Jones, K.: What is the role of NLP in text retrieval? [20] 1–24

[3] Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural Language Interfaces to Databases – An Introduction. Journal of Natural Language Engineering **1** (1995) 29–81

[4] Copestake, A., Jones, K.S.: Natural Language Interfaces to Databases. The Knowledge Engineering Review **5** (1990) 225–249

[5] Perrault, C., Grosz, B.: Natural Language Interfaces. Exploring Articial Intelligence (1988) 133–172

[6] Fuhr, N., Großjohann, K.: XIRQL: A Query Language for Information Retrieval in XML Documents. In Croft, W., Harper, D., Kraft, D., Zobel, J., eds.: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York City, NY, USA, ACM Press, New York City, NY, USA (2001) 172–180

[7] Sigurbjörnsson, B., Trotman, A., Geva, S., Lalmas, M., Larsen, B., Malik, S.: INEX 2005 Guidelines for Topic Development (2005) http://inex.is.informatik.uni-duisburg.de/2005/internal/pdf/TD05.pdf.

[8] Trotman, A., Sigurbjrnsson, B.: Narrowed Extended XPath I (NEXI). [21] 16–40

[9] Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: International Conference on New Methods in Language Processing. (1994)

[10] Tannier, X., Girardot, J.J., Mathieu, M.: Analysing Natural Language Queries at INEX 2004. [21] 395–409

[11] Arampatzis, A., van der Weide, T., Koster, C., van Bommel, P.: Linguistically-motivated Information Retrieval. In Kent, A., ed.: Encyclopedia of Library and Information Science. Volume 69. Marcel Dekker, Inc., New York, Basel (2000) 201–222

[12] Moreau, F., Sbillot, P.: Contributions des techniques du traitement automatique des langues la recherche d'information. Technical Report 1690, IRISA, France (2005)

[13] Tzoukermann, E., Klavans, J.L., Jacquemin, C.: Effective use of natural language processing techniques for automatic conflation of multi-word terms: the role of derivational morphology, part of speech tagging, and shallow parsing. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, USA, ACM Press, New York City, NY, USA (1997) 148–155

[14] Arampatzis, A.T., Tsoris, T., Koster, C.H.A., van der Weide, T.P.: Phrase-based Information Retrieval. Information Processing & Management **34** (1998) 693–707

[15] Fabre, C., Jacquemin, C.: Boosting Variant Recognition with Light Semantics. In: Proceedings of the 18th International Conference on Computational Linguistics, COLING 2000, Saarbrcken (2000) 264–270

[16] Kazai, G., Lalmas, M.: INEX 2005 Evaluation Metrics (2005) http://inex.is.informatik.uni-duisburg.de/2005/inex-2005-metricsv4.pdf.

[17] Geva, S., Leo-Spork, M.: XPath Inverted File for Information Retrieval. In Fuhr, N., Lalmas, M., Malik, S., eds.: Proceedings of the second Workshop of the Initiative for the Evaluation of XML retrieval (INEX), December 15–17, 2003, Schloss Dagstuhl, Germany (2004) 110–117

[18] Sauvagnat, K., Boughanem, M., Chrisment, C.: Searching XML documents using relevance propagation. In: String Processing and Information Retrieval, Padoue, Italy, Springer-Verlag, New York City, NY, USA (2004) 242–254

[19] Woodley, A., Geva, S.: NLPX at INEX 2004. [21] 382–394

[20] Strzalkowski, T., ed.: Natural Language Information Retrieval. Kluwer Academic Publisher, Dordrecht, NL (1999)

[21] Fuhr, N., Lalmas, M., Malik, S., Szlàvik, Z., eds.: Advances in XML Information Retrieval. Third Workshop of the Initiative for the Evaluation of XML retrieval (INEX). In Fuhr, N., Lalmas, M., Malik, S., Szlàvik, Z., eds.: Advances in XML

Information Retrieval. Third Workshop of the Initiative for the Evaluation of XML retrieval (INEX). Volume 3493 of Lecture Notes in Computer Science., Schloss Dagstuhl, Germany, December 6-8, 2004, Springer-Verlag, New York City, NY, USA (2005)