Natural Language Processing and XML Retrieval

Alan Woodley
Faculty of Information Technology
Queensland University of Technology
awoodley@qut.edu.au

Xavier Tannier
Dept. Networks, Information, Multi-media
Ecole des Mines Saint-Etienne
tannier@emse.fr

Marcus Hassler
Department of Informatics
Universitdat Klagenfurt
Marcus.Hassler@uni-klu.ac.at

Shlomo Geva
Faculty of Information Technology
Queensland University of Technology
s.geva@qut.edu.au

Abstract

XML information retrieval (XML-IR) systems respond to user queries with results more specific than documents. XML-IR queries contain both content and structural requirements traditionally expressed in a formal language. However, an intuitive alternative is natural language queries (NLQs). Here, we discuss three approaches for handling NLQs in an XML-IR system that are comparable to, and even outperform formal language queries.

1 Introduction

Information retrieval (IR) systems respond to user queries with a ranked list of relevant documents, even though only parts of the documents are relevant. In contrast, XML-IR systems are able to exploit the separation of structure and content in XML documents by returning relevant *portions* of documents. To interact with XML-IR systems users must specify both their content and structural requirements in structured queries. Currently, formal languages are used to specify structured queries, however, they have proven problematic since they are too difficult to use and are too tightly bound to the collection.

A promising alternative to formal queries languages is structured natural language queries (NLQs). Here, we present justifications for NLQs in XML-IR, and describe three approaches that translate NLQs to an existing formal language (NEXI). When used in with an XML-IR system the approaches perform strongly, at times outperforming a baseline consisting of manually constructed NEXI expressions. These results show that NLQs are potentially a viable alternative to XML-IR systems.

2 Motivation

There are two major problems with formal query languages for XML-IR that could be rectified with NLQs. First, expressing a structural information need in a formal language is too difficult for many users. O'Keefe and Trotman (2004) investigated five structured query languages and concluded that all of them were too complicated to use. In practice, 63% of the expert-built queries queries in the 2003 INEX campaign had major semantic or syntactic errors, requiring up to 12 rounds of corrections. In contrast, users should be able to express their need in NLQs intuitively.

Second, formal query languages require an intimate knowledge of a document's structure. So, in order to retrieve information from abstracts, sections or bibliographic items, users need to know their corresponding tags. While this information is contained in the DTD/ Schema, it may not be publicly available, and is too much information to remember (INEX, for instance has 192 nodes). The problem extrapolates in a heterogenous collection since a single retrieval unit could be expressed in multiple tags. In contrast, since structures in NLQs are formulated at the conceptional level users do not have to know their actual tag names.

3 The Approaches

Here, we present three techniques used to translate NLQs to NEXI in INEX 2004 and 2005. The three approaches are called *Hassler*, *Tannier* (Tannier, 2005) and *Woodley* (Woodley and Geva, 2005) after their authors. While each of the approaches is different, they all contain four main stages.

3.1 Detecting Structural and Content Constraints

The first stage is to detect a query's structural and content constraints. *Hassler* uses template matching based on words and parts-of-speech. Links between structure and content are not linguistically motivated, and it is assumed that content is the last element. *Woodley* adds shallow syntactic parsing before applying the same kind of template matching. *Tannier* uses deep syntactic analysis, complemented by some specific semantic rules concerning query structure.

3.2 Structure Analysis

The second stage is to map structural constraints to corresponding XML tags. This requires lexical knowledge about the documents' structure, since the tags in the XML documents are rarely "real" words or phrases, but abbreviations, acronyms or an amalgamation of two. Furthermore, a single tag can be referred to by different names. *Tannier* uses grammatical knowledge to recognise some frequent linguistic constructions that imply structure.

3.3 Content Analysis

The third stage is to derive users' content requirements, as either terms or phrases. Noun phrases are particularly useful in information retrieval. They are identified as specific sequences of parts-of-speech. *Tannier* is also able to use content terms to set up a contextual search along the entire structure of the documents.

3.4 NEXI Query Formulation

The final stage of translation is the formulation of NEXI queries. Following NEXI format, content terms are delimitated by spaces, with phrases surrounded by quotation marks.

4 Results

Here, we present the ep-gr scores from the 2005 INEX NLQ2NEXI Track. The results correspond to different relevance quantisation and interpretations of structural constraints - a thorough description of which is provided in (Kazai and Lalmas, 2005). The results compare the retrieval performance of a XML-IR system (Geva, 2005) when the 3 natural language approaches and a fourth "baseline" system, which used manually constructed NEXIs queries, were used as input.

	Baseline	*Hassler*	*Tannier*	*Woodley*
Strict	**0.0770**	0.0740	0.0775	0.0755
Gen	**0.1324**	0.1531	0.1064	0.1051

Table 1: SSCAS ep-gr scores

	Baseline	*Hassler*	*Tannier*	*Woodley*
Strict	**0.0274**	0.0267	0.0304	0.0267
Gen	**0.0272**	0.0287	0.0298	0.0311

Table 2: SVCAS ep-gr scores

	Baseline	*Hassler*	*Tannier*	*Woodley*
Strict	**0.0383**	0.0338	0.0363	0.0340
Gen	**0.0608**	0.0641	0.0682	0.0632

Table 3: VSCAS ep-gr scores

	Baseline	*Hassler*	*Tannier*	*Woodley*
Strict	**0.0454**	0.0372	0.0418	0.0483
Gen	**0.0694**	0.0740	0.0799	0.0742

Table 4: VVCAS ep-gr scores

The results show that the NLP approaches perform comparably - and even outperform - the baseline.

5 Conclusion

While the application of NLP XML-IR is in its infancy, it has already produced promising results. But if it is to process to an operational environment it requires an intuitive interface. Here, we describe and presented the performance of three approaches for handling NLQs. The results show that NLQs are potentially a viable alternative to formal query languages and the integration of NLP and XML-IR can be mutually beneficial.

References

Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltàn Szlàvik, editors. 2006. *INEX 2005 Proceedings, Schloss Dagstuhl, Germany, November 28–30, 2005.*

Shlomo Geva. 2005. GPX - Gardens Point XML Information Retrieval at INEX 2005. In Fuhr et al. (Fuhr et al., 2006), pages 211–223.

Gabriella Kazai and Mounia Lalmas. 2005. INEX 2005 Evaluation Metrics. http://inex.is.informatik.uni-duisburg.de/2005/inex-2005-metricsv4.pdf.

Richard A. O'Keefe and Andrew Trotman. 2004. The Simplest Query Language That Could Possibly Work. In *Proceedings of the second Workshop of the INEX, Schloss Dagstuhl, Germany.*

Xavier Tannier. 2005. From Natural Language to NEXI, an interface for INEX 2005 queries. In INEX05 (Fuhr et al., 2006), pages 289–313.

Alan Woodley and Shlomo Geva. 2005. NLPX at INEX 2005. In INEX05 (Fuhr et al., 2006), pages 274–288.