

QXtract: A Building Block for Efficient Information Extraction from Text Databases

Eugene Agichtein Luis Gravano

Columbia University
{eugene, gravano}@cs.columbia.edu

Background: A wealth of information is hidden within unstructured text. This information is often best utilized in structured or relational form, which is suited for sophisticated query processing, for integration with relational databases, and for data mining. For example, newspaper and e-mail archives contain information that could be useful to analysts and government agencies. Information extraction systems produce a structured representation of the information that is “buried” in text documents. Unfortunately, processing each document is computationally expensive, and is not feasible for large text databases or for the web. With many database sizes exceeding millions of documents, processing time is becoming a bottleneck for exploiting information extraction technology.

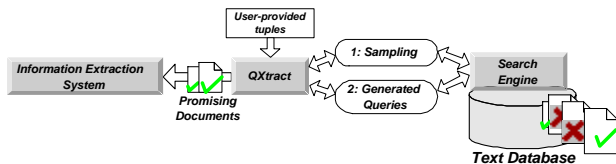


Figure 1: The architecture of an efficient information extraction system that identifies promising documents via querying.

Recently, we presented the *QXtract* system [1], which automatically generates queries to identify the promising database documents for extraction by an arbitrary information extraction system. By focusing only on potentially useful documents and ignoring the rest, we can dramatically improve the efficiency and scalability of information extraction. *QXtract* (Figure 1) discovers the characteristics of documents that are useful for extraction of a target relation by sampling the database with tuples for the relation. This document sample is then processed by the information extraction system of choice, resulting in an automatically “labeled” training sample of “useful” and “useless” documents. Machine learning and information retrieval techniques are then used to generate queries for retrieving additional useful documents that are in turn processed by the information extraction system to extract the final relation.

Demo Interface and Operation: We demonstrate a practical and efficient information extraction architecture based on *QXtract*. Extracting a user-defined relation using our system involves three stages: task specification, *QXtract* training, and the final extraction of the target relation. Our prototype can incorporate user feedback during all stages of the process.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD 2003, June 9-12, 2003, San Diego, CA.

Copyright 2003 ACM 1-58113-634-X/03/06 ...\$5.00.

Task Specification: The user chooses one of the available information extraction systems, the target relation to extract and a handful of example tuples for the relation, and the text database of interest.

QXtract Training: During the training process outlined above the system displays intermediate results (e.g., the current set of extracted tuples), and indicates overall progress.

Promising Document Retrieval and Final Extraction: In the final stage of the process, the tuples are displayed as they are extracted, as well as relevant statistics including estimated completeness of the extracted relation.

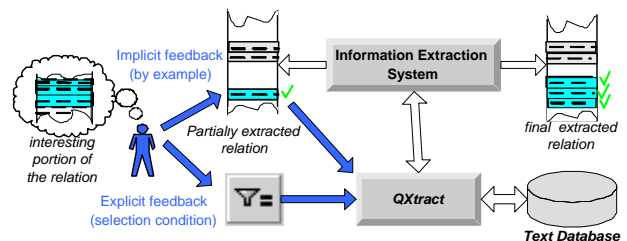


Figure 2: Interactive information extraction from text databases.

User Feedback: Throughout the process the user is able to adjust the retrieval parameters, to correct the system (e.g., by indicating invalid extracted tuples), and to interrupt or re-start the process with different *QXtract* parameters or different seed tuples. Additionally, our interface allows the user to focus the extraction process on the most “interesting” portion of the target relation, as shown in Figure 2. The system supports two ways to identify the interesting portion of the relation:

- **Implicit (by example):** The user marks some of the extracted tuples as “interesting.” Based on this input, *QXtract* attempts to characterize the contexts in the documents where these tuples occurred and thus generate more specific queries.

- **Explicit (selection condition):** The user augments (restricts) the queries by specifying string-based selection conditions that explicitly describe the interesting portions of the relation (e.g., by providing keywords that match the attributes of “interesting” tuples). These conditions are “pushed down” into *QXtract* queries.

In summary, our system demonstrates a novel interactive end-to-end information extraction architecture. The techniques showcased therein can be used as a building block for scalable, efficient, and effective information extraction from text databases. Please refer to <http://snowball.cs.columbia.edu> for more information about the underlying technology and this demo.

ACKNOWLEDGEMENTS: This research is supported by the National Science Foundation under Grants No. IIS-97-33880 and IIS-98-17434.

[1] E. Agichtein and L. Gravano. Querying text databases for efficient information extraction. *Proceedings of the 19th IEEE International Conference on Data Engineering (ICDE)*, 2003.