A Dynamic Interface for a Dynamic Task: Information Extraction and Evolving Queries

Peter Vanderheyden and **Robin Cohen**
Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1
{*pbvander, rcohen*}*@uwaterloo.ca*

Abstract

Information gathering systems do not typically find all the information desired by the user on their first try, and so a cycle of query refinement occurs. Information retrieval systems — which classify documents as either relevant or irrelevant to a user's query — allow the user to refine a query either directly by changing the wording of the query, or indirectly by examining search results and accepting or rejecting documents, integrating these selections into the query as positive and negative search terms. Studies of how library patrons interacted with (human) librarians suggest that queries evolving over time is a natural component of the information gathering process (*e.g.*, (Bates forthcoming), (Twidale & Nichols 1998)). Query refinement or evolution is not well supported, however, in many of the current *information extraction* (IE) systems — which locate specific query-relevant information in documents and present this information to the user. It is our belief that the common "naive" user would benefit from a more flexible cycle of interaction and query refinement during information extraction, a cycle in which a shared representation for system knowledge would enable the user and system to negotiate their roles and vary their levels of initiative within a given task (Vanderheyden & Cohen 1998).

Information gathering tasks such as information extraction (Cowie & Lehnert 1996), as well as information retrieval and information filtering (Belkin & Croft 1992), are inherently interactive tasks. That is, they cannot be performed without some human intervention, because they attempt to model and satisfy the demands of a human user. Firstly for any such task, fully autonomous approaches are far from being able to accurately decide whether the information found by the system reflects a user's subjective criteria of relevance. Secondly for information extraction specifically, it is difficult if not impossible to predict how an item of interest will be expressed in any unstructured text, so that a computational approach could be sure to identify it. While the interactive nature of information retrieval has been considered both in human-human and human-computer situations (as we will discuss below), little has been said with reference to information extraction about the role of interaction and the degree of system autonomy. It is our view that interaction between user and system is a basic and integral component in any natural approach to information extraction, and will be a growing area of interest.

Briefly, the process of performing an information extraction task involves the user (either the end-user, or perhaps an expert user such as the system developer) annotating the text of several documents in the corpus, and from these annotations the system learns by induction the rules for annotating subsequent text. Typically, annotations are in the form of SGML markups; they can indicate general linguistic information (*e.g.*, part of speech, or semantic category) as well as domain-specific information, and a subset is used to identify the specific information requested in the query. The user then directs the system to apply these rules to novel corpus documents. After reviewing the system's annotations, the user either accepts the query elements identified by the system in the text or modifies the system's annotation rules (by manually modifying the rules directly, or by modifying the annotations and then having the system learn new rules from them). This train-and-test cycle continues until the user is satisfied with the results and no more documents remain to be examined.

As a running example, we will consider the information extraction task of identifying instances of layoffs that have occurred in the past six months as reported in a corpus of newspaper articles, and expressing the results as a table with one entry for each such instance containing the name of the employer, the number of employees laid off, and the current status of the layoff — completed, ongoing, or planned (Figure 1, top). Each instance of a layoff event can be expressed as a *template* structure (Figure 1, bottom), with the employer and employee elements referring in turn to subordinate template structures containing elements of their own, including for example the name of the employer and the number of employees laid off, respectively[1].

It would be ideal to be able to examine the kinds of interactions that occur in actual IE situations. As this data is not available, we begin by considering the in-

	employer-name	employee-num	status
layoff 1			
layoff 2			
⋮			

```
layoff-event:
  employer-entity: ...
  employee-entity: ...
  status-atomic:  {completed,ongoing,planned}
```

Figure 1: Two views of the information extraction task: filling a table with one layoff event in each row (top); instantiating a "template" of the layoff event, and filling in its elements (bottom).

teractions in another information gathering task, information retrieval, in either the classic sense of the term — users interacting with a (human) librarian in search of relevant documents in a library — as well as in its modern sense — users interacting with a computerised facility for searching an on-line corpus of documents. Studies of the former (Bates forthcoming) have shown that librarians and patrons are often involved in periods of negotiation that are not directly relevant to finding the desired documents, and that library patrons often do not initially convey exactly what they're looking for perhaps until they have established an understanding of a common context with the librarian. Also, the search itself may change with time as the patrons re-evaluate exactly what information they are looking for. Similar findings have been discussed in the context of online searches using the ARIADNE system (Twidale & Nichols 1998). The TREC series of conferences on information retrieval — in which participants "compete" to find the highest proportion of documents relevant to various queries — likewise acknowledges the interactive nature of information retrieval, recently establishing a special track in which participants develop their queries interactively with their systems (*e.g.*, (Clarke & Cormack 1996)). In this way, users can find effective search terms and refine their queries.

Turning from information retrieval to information extraction, the task seems inherently to be even more interactive[2]. In most current information extraction systems, however, the user has full control (*i.e.*, maintains the *initiative*) over when the system begins to process the text and when it should end, and while some interaction with the system is assumed, it is often quite limited. We can classify the interaction and degree of system autonomy during this process as falling into one of several categories:

(i) no autonomy (*e.g.*, FASTUS (Appelt *et al.* 1995)) — the user performs all aspects of the information extraction task, with the system simply acting as an interface to the corpus and applying whatever annotation rules have been represented in propositional form by the user;

(ii) partial with absolute parameters — the user could specify, for example, that the system should operate autonomously for a fixed period (*e.g.*, for X seconds, or until it has processed Y documents); this assumes that the user knows ahead of time (or is not concerned with) characteristics of the corpus that affect such things as the time requirements and accuracy of results, or that those characteristics are fairly constant — a conclusion for which there is not yet any evidence[3] — whereas intuitively one might expect some tasks to be easier than others, and requiring differing amounts of user involvement;

(iii) partial with relative parameters (*e.g.*, Alembic Workbench (Day *et al.* 1997), UNO (Iwańska *et al.* 1995)) — the user gives up the initiative and the system operates autonomously until some relative "stop criterion", such as a given level of uncertainty, has been reached, regardless of whether this will require processing one document or one thousand[4]; this approach is more flexible than (i) and (ii);

(iv) negotiable, or *mixed initiative* (*c.f.*, (Allen 1994)) — the exact roles of system and user are not predetermined, and either can interrupt the other; this is

[1] Conceptually, we believe that the information extraction task can be divided into three subtasks:

- developing a model of the domain of inquiry (*domain model*), in order to identify contexts relevant to the query for which templates should be instantiated;
- developing a model of the specific information requested in the query (*query model*), containing some subset of the domain model possibly with further constraints, with which to fill a template once it has been instantiated;
- developing a model of how the domain of inquiry is expressed in the corpus of texts being processed (*corpus model*), in order to map from the text to elements in the domain and query models.

This tripartite structure influences how we look at the role of user-system interaction in an information extraction system; further details are as yet forthcoming.

[2] Whereas it may be possible to process some information retrieval queries reasonably well using only autonomous means (*e.g.*, searching on the basis of keywords and related words found in an on-line thesaurus, then correcting for term frequencies), even a simple information extraction task requires a degree of natural language understanding and domain knowledge — the domain knowledge to know what it's looking for, and the NLU to be able to identify it in the text — that a system could not presently perform at all well without human intervention.

[3] We have not yet run across any studies of users performing information extraction, in which the frequency and effects of variations according to query, user, domain, *etc.*, have been investigated.

[4] Combining relative and absolute parameters should not be difficult, and may in fact already be supported on these systems.

quite similar to (iii) with interruptibility, except that it makes explicit the need for a shared representation of the current state of the information extraction task, accessible and understandable to both system and user;

(v) full autonomy — this will only be possible, for the general case, when the "AI-complete" problems of natural language understanding and automatic user modelling have been solved.

A number of systems provide graphical interfaces for the annotation phase in order to accelerate and organise the process of manually editing text in order to add annotations (including, *e.g.*, Alembic Workbench (Day *et al.* 1997), FASTUS (Appelt *et al.* 1995), and the system by Bagga *et al.* (Bagga 1997)). Thus, the job of annotating text with a given set of annotation types — of using world knowledge in order to recognise relevant entities and events in the text and marking them — is acknowledged as one that the user must play a part in, and that a graphical user interface can support nicely. This still leaves open the question of what template entries, and therefore annotation types, are appropriate for a given query and domain, a problem acknowledged as difficult (Onyshkevych 1993)[5] but for which the current interfaces offer no assistance. The reader may recall for example the information retrieval strategies by library patrons mentioned earlier, where queries evolve through interactions with the librarian and examination of documents found by earlier iterations of the information gathering process.

In the later stage of the information extraction task, when the system induces rules from the annotated text, current systems give little support. In current systems, rules are displayed to the user in the same propositional language in which they are represented in the system (*e.g.*, the FASTSPEC rule specification language, for FASTUS), and manually modifying these rules represented in this way is not an easy task (Appelt *et al.* 1995). However, this kind of rule representation can be very effective for the processing performed by the system, in which it is important (in some systems) to learn these rules automatically and (in all systems) to maintain the consistency of a large number of rules efficiently.

In short, information extraction in the everyday offers a number of challenges that are not yet being addressed in current systems. For a novel query and domain of interest, how does the user decide on the formalisation of the query as a template, and on the elements of the text that need to be annotated? If the natural course of information gathering is for the query to evolve, then these query elements and annotations may well change; it would be helpful to support this evolutionary process without requiring the user to begin a new information extraction task from scratch. As annotation rules are developed, is there a more intuitive and "user-friendly" form than to have the user examine and modify a propositional representation of those rules? For many people, the rules of syntax for natural language are often unintuitive, unfamiliar, and possibly unknown; a propositional representation of rules for a domain-specific sublanguage would be equally or more unwieldy. When applying those rules to novel text brings up instantiations of the query that contain partial or incorrect information, what kind of support can the system provide for improving the accuracy on the next iteration of the extraction task? If preliminary results are not to the user's satisfaction — if the system returns templates that are only partially filled, or contain incorrect information, or fails to return templates when they are called for — it would be helpful to be able to recognise which rules are associated with particular results [6].

In conclusion, we leave the reader with a number of possible information extraction scenarios (within the layoff query domain) to consider, as well as how a system might handle them given extended capabilities for user interaction and operational autonomy:

- a system misinterprets the meaning of a sentence (*e.g.*, taking "[a person within some organisation] agreed to lay off..." to refer to a corporate layoff, whereas 'lay off' was in fact used in the sense of 'to stop doing or taking something' (Webster's Dictionary)), and returns template instantiations containing incorrect information; when the user consults the text and rejects the incorrectly filled template, the system takes the initiative to engage in a clarification dialogue with the user in order to isolate the sentence forms that suggest this sense of the term, optionally indicating examples of the sentence pattern that led to the error;
- in a similar situation to the one above, the system takes the initiative to indicate to the user several rules that are highly ambiguous, possibly using examples from the text to illustrate the ambiguity; in this way, clarification requests could be kept to a minimum so as not to annoy the user[7];
- on the basis of preliminary system results, a user realises that layoffs in which employees have been re-

[5] Onyshkevych (1993) writes: "The design of the template needs to balance a number of (often conflicting) goals, as reflected by these desiderata...": descriptive adequacy, clarity, determinacy, perspicuity, monotonicity, application considerations, and reusability (p. 141).

[6] In some situations, it may be the case that the user is going to examine all the documents eventually anyway; then the importance of perfect accuracy may be less. For example, the MITA system (Glasgow *et al.* 1997) was designed for the specific application of assisting insurance actuaries to review insurance claims, and all of the information it returns is examined by a human actuary — results with a low confidence rating are examined in depth, while templates filled with a higher confidence may be examined only superficially. One might imagine that in most cases, however, the user will not be interested in reviewing all the documents in the corpus, relevant and irrelevant alike.

[7] This kind of "sample selection" technique is used in various learning contexts; *e.g.*, (Engelson & Dagan 1996).

called should be discounted, and adds new annotations in order to identify recall events; the user enters into a clarification dialogue with the system in order to modify the system's representation of the layoff domain and the system subsequently suggests appropriate annotations that will need to be added;

- in a similar situation to the one above, the system takes the initiative to include in its domain representation possible distractors in the corpus — that is, terms that appear in contexts similar to the query elements and that may or may not be relevant to the query[8] — and the user is able to manipulate these terms within the shared representation of the query domain, as well as add or delete them.

It may appear that these suggestions are of concern only to a user who is new to the system, to the domain, or to the corpus, as this unfamiliarity would easily lead to inefficient use of the information extraction system (Vanderheyden & Cohen 1998). It is often the case, however, that system capabilities designed to assist new users are found to be equally helpful, or even more so, for experienced users undertaking more complex tasks. The role of interaction between the user and the information extraction system needs to be examined more closely; systems with a greater potential for autonomy can adapt and become more supportive and effective at their task.

References

Ahonen, H.; Heinonen, O.; Klemettinen, M.; and Verkamo, A. I. 1997. Mining in the phrasal frontier. Technical Report C-1997-14, Department of Computer Science, University of Helsinki.

Allen, J. F. 1994. Mixed initiative planning: Position paper. Presented at the ARPA/Rome Labs Planning Initiative Workshop.

Appelt, D. E.; Hobbs, J. R.; Bear, J.; Israel, D.; Kameyama, M.; Kehler, A.; Martin, D.; Myers, K.; and Tyson, M. 1995. SRI International FASTUS system: MUC-6 test results and analysis. In MUC-6 (1995), 237-248.

Bagga, A. 1997. The role of a GUI in the creation of a trainable message understanding system. In Johnson, J. H., ed., *Proceedings of CASCON '97: Meeting of Minds*, 261-271.

Bates, M. J. forthcoming. Indexing and access for digital libraries and the internet: Human, database, and domain factors. *Journal of the American Society for Information Science.*

Belkin, N. J., and Croft, W. B. 1992. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM* 35(12):29–38.

Clarke, C. L. A., and Cormack, G. V. 1996. Interactive substring retrieval (MultiText experiments for TREC-5). In *Proceedings of TREC-5.* TREC.

Cowie, J., and Lehnert, W. 1996. Information extraction. *Communications of the ACM* 39(1):80–91.

Day, D.; Aberdeen, J.; Hirschman, L.; Kozierok, R.; Robinson, P.; and Vilain, M. 1997. Mixed-initiative development of language processing systems. In *Fifth Conference on Applied Natural Language Processing: Proceedings of the Conference.* Washington, D.C., USA: ACL.

Engelson, S. P., and Dagan, I. 1996. Sample selection in natural language learning. In Wermter, S.; Riloff, E.; and Scheler, G., eds., *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence. Springer, Berlin. 230–245.

Glasgow, B.; Mandell, A.; Binney, D.; Ghemri, L.; and Fisher, D. 1997. MITA: An information extraction approach to analysis of free-form text in life insurance applications. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 97) and the 9th Conference on Innovative Applications of Artificial Intelligence (IAAI 97)*, 992–999. Providence, Rhode Island, USA: AAAI.

Iwańska, L.; Croll, M.; Yoon, T.; and Adams, M. 1995. Wayne State University: Description of the UNO natural language processing system as used for MUC-6. In MUC-6 (1995), 263–277.

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland, USA: Morgan Kaufmann: San Francisco, USA.

Onyshkevych, B. 1993. Template design for information extraction. In *TIPSTER Text Program Phase I: Workshop Proceedings*, 141–145. Fredricksburg, Virginia, USA: TIPSTER.

Riloff, E. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 811–816. AAAI.

Twidale, M. B., and Nichols, D. M. 1998. Designing interfaces to support collaboration in information retrieval. *Interacting With Computers* 10(2):177–193.

Vanderheyden, P., and Cohen, R. 1998. Information extraction and the casual user. In *Proceedings of the AAAI'98 Workshop on AI and Information Integration*, 137–142. Madison, Wisconsin, USA: AAAI.

[8] Techniques developed within the area of knowledge discovery, or data mining, may be applicable here (*e.g.*, (Ahonen *et al.* 1997)) for identifying terms of possible relevance. Then techniques such as the lexical acquisition methods used in the AutoSlog system (Riloff 1993) might be used to integrate the new terms into the domain representation.