Information Extraction and the Casual User

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

Abstract

The task of information extraction (IE) calls for a limited understanding of text, limited by the demands of the user and the domain of inquiry — IE returns specific instances of the concept or relation of interest to the user. Whereas IE systems have, to date, been oriented towards either system experts (*e.g.*, computational linguists) or domain experts (*e.g.*, professionals searching for information within the field of their profession), the availability of large amounts of on-line textual information to the casual user strongly suggests that techniques oriented towards non-experts are needed. We present a review of current user-involvement techniques in IE, and begin to investigate issues of knowledge representation and learning in the development of a mixed-initiative information extraction system. In particular, we discuss some advantages of dividing the knowledge used by the IE system into a query model, a domain model and a corpus model, to assist casual users in interacting with the system. We also advocate flexibility in determining increments for learning, supporting negotiation between system and user.

Introduction

Given the large amount of on-line text available today, it is no surprise that a number of approaches have arisen to assist us in making sense of it all. Information retrieval (IR) — evaluating the relevance of a text to a user — is one example, and information extraction (IE) is another. Information extraction (Cowie & Lehnert 1996) is interested in finding not only *whether* a text contains information relevant to a user's request, but specifically *what* that relevant information is and how it relates to the request. For example, a user searching a corpus of on-line news articles or World Wide Web pages and interested in job losses might pose the IR query "Find 100 documents relevant to layoffs and job losses, and return them in order of relevance", and follow up with the IE query "In those 100 documents, identify the specific instances of who was laid off by what company".

Information extraction can be thought of as a limited form of natural language understanding. While it is often difficult to determine when a machine truly "understands" a text, an IE task is complete when all instances of the specified text patterns have been identified.

Reading and analyzing large amounts of text is at once a demanding and a tedious task, requiring a great deal of time and attention to detail. Even trained and practised human analysts make a considerable number of mistakes (*e.g.*, (Okurowski 1993) measured an error rate of 30%). The appeal of automating this task is clear — computers are fast and immune to the effects of tedium. However, full automation is akin to full natural language understanding by machines, a still evasive goal, and therefore some user involvement is necessary in information extraction. An interesting question, then, is what shape should this involvement take?

Whereas most IE systems have, to date, been oriented towards either system experts or domain experts, the availability of large amounts of on-line textual information to the casual user strongly suggests that techniques oriented towards non-experts are needed. Current IE systems require the user to design and understand a sizable body of text analysis rules based either on formal linguistic or *ad hoc* pattern-matching approaches, and to have a well-developed and formalized understanding of the domain of inquiry. Furthermore, adapting an IE system to a new domain requires several man-days or even man-months of work to reach performance levels of between 50-60%. In contrast, many computer users may not have the time, expertise, or inclination that these approaches demand.

A typical IE task

It is often much easier to present a discussion in the context of a specific example, and we will refer to the example described here throughout this paper. An information extraction task typically involves a user formulating a query that contains a slot for each piece of information to be extracted from the text. The output from the system will be a series of instantiations of this query with slots filled, or left blank if the information was not available in the document. Rules need to be defined that map patterns in the text into fillers for output slots.

A user might be interested in the following:

> "Identify all companies that have laid off workers in the past six months."

and formalizes this request in the form of a query template:

```
Layoff-Company:
Layoff-EmployeesAffected:
Layoff-StartDate:
```

Finding the following paragraph in a newspaper article:

> This city's economic woes continued Tuesday with the plant closure announcement by Kershaw Manufacturing Canada Ltd., resulting in the layoff of 17 people.[1]

the system would process it using lexical information and word pattern rules in its knowledge base. These rules would need to:

- recognize that the text contains information about a layoff event — in this case the word "layoff" is a good indicator, and a `Layoff` template is (tentatively) instantiated;
- identify important terms — the "Ltd." in "Kershaw Manufacturing Canada Ltd." identifies it as a company name, and a `Company` term is instantiated;
- map terms to the query elements — the single instance of `Company` in the context of `Layoff` is selected to fill the `Layoff-Company` slot, and the structure of the phrase "the layoff of 17 people" identifies "17 people" as the filler for the `Layoff-EmployeesAffected` slot.

Finally, at the end of this process, the system returns the filled template:

```
Layoff-Company:
   Kershaw Manufacturing Canada Ltd.
Layoff-EmployeesAffected:
   17
Layoff-StartDate:
   Tuesday, <date>
```

with `<date>` derived from the date of the newspaper article.

The casual user

We have already mentioned that the end-user has an inevitable role in guiding an information extraction task, and that current systems do not address some of the issues involved. Before continuing, it might be helpful to provide a clearer picture of what we mean by a "casual" user:

- not a system expert — a user who is neither a system developer nor a computational linguist, and who may have little *a priori* knowledge of formal linguistics or of how sentences are handled by various modules of the system;
- not a domain expert — a user may pose a query about an unfamiliar domain, or "on-the-fly" without having considered all of the possible relevant concepts;
- modest time and effort available — we propose the very ambitious goal that a user should be able to obtain a satisfactory solution from the system, using an untagged text corpus (*e.g.*, a corpus of newspaper articles), in less than one man-day and with as intuitive and efficient an interface as possible.

To illustrate potential difficulties that such a user might have, we consider again the request:

> "Identify all companies that have laid off workers in the past six months."

Immediately this raises issues of how to interpret the user's terminology in order to formalize the query for an information extraction task:

1. what does "identify" mean? — is it sufficient to identify a company by reporting its name?
2. what does "company" mean? — is a corporation a "company"? is a government agency?
3. what does "lay off" mean? — does it include firing, quitting, retiring, all company closures? what if the workers have been hired elsewhere, or recalled, or the layoff has been cancelled?
4. what does "worker" mean? — does this include all employees, including high-ranking officials, part-time employees, summer students?
5. what does "the past six months" mean? — is it six months from this day (*e.g.*, if this is August 15, then since February 15), or this month (*e.g.*, since February 1)?
6. how should uncertain or incomplete information be reported, if at all?

In the Message Understanding Conferences (MUCs, the main venue for reporting work in IE) and in other large-scale applications, many of these questions are answered with the help of experts (*e.g.*, consulting trained judges in MUC-5 (Will 1993); consulting accountants for the extraction of financial information by MITA (Glasgow *et al.* 1997); consulting nurses for extraction of medical information by CRYSTAL (Soderland *et al.* 1995); constraining the language only to legal documents in LegalDocs (Holowczak & Adam 1997)). However, such resources will not always be available.

A role for mixed-initiative

It is our opinion that the investigation of techniques oriented to casual users will be closely linked to a more collaborative, mixed-initiative framework for representation of system knowledge and exchange of information with the user. A mixed-initiative strategy is one where each party contributes to a task what it does best, all parties negotiating their respective roles dynamically during the course of the task (Allen 1994). In the context of IE, a user could use knowledge about

[1] This paragraph is taken from the Ottawa Citizen newspaper (July 11, 1990).

the domain of inquiry to steer the user-system team in a top-down fashion, while the system examines word patterns in the text to contribute in a bottom-up fashion. These roles would change to handle problems as they arise; for example, the user may correct the system when it incorrectly analyzes some construct in the text. At least one other system has already begun to integrate mixed-initiative into IE (Alembic Workbench (Day *et al.* 1997)), and differences between their approach and ours will be discussed in the section on our proposed system.

One factor that stands in the way of applying a mixed-initiative in IE is the problem of system autonomy — a system that relies entirely on the user has no basis for negotiating a greater role for itself. However, learning enables a system to automatically extend or refine its knowledge base. A number of current IE systems use learning algorithms to infer new knowledge on their own, rather than the user being solely responsible for integrating new knowledge into the system's representation. This in turn can support a mixed-initiative strategy, giving the system the potential for greater control while reducing the time demands on the casual user.

As well, communication between system and user is necessary to ensure that what the system learns is consistent with what the user is interested in. Therefore, another important consideration is how system knowledge is represented, in order to support effective communication.

To address the topic of the workshop, namely how to accommodate the current explosion in information that is available on-line, our point of view is that information extraction is one procedure which will be of interest to users and that information extraction can be designed to work more effectively with casual users. What is needed is a facility for allowing users to query the system and then for the system and user to interact, with the system developing independent learning techniques, so that there is not too much expected of the user. Below we discuss how the balance of responsibilities between the user and system can be achieved. This addresses the workshop subtopic of approaches for query planning, necessary for access to information.

Overview of current systems

Many current systems divide the IE task into stages, modeled either on traditional deep-semantic natural language processing — part-of-speech tagging, full-sentence syntactic parsing, semantic and pragmatic analysis (Morgan *et al.* 1995) — or on shallower-semantic pattern matching (Appelt *et al.* 1995), or on a combination of the two (*e.g.*, (Weischedel 1995), (Fisher *et al.* 1995), and a number of others). Text is read by the first stage and analyzed, then the results of that analysis are passed with the text to the second stage, and so on, until the text has passed through each stage. Stages are often developed and (if they involve learning) trained in this sequential manner as well, with the developer first concentrating on one stage then, when the output of that stage is mostly correct, working on the next.

While some system knowledge may be centralized, more often it is distributed across and specialized to the different processing stages. This includes processing rules — rules for performing part-of-speech tagging are localized to one stage, while rules for syntactic parsing are contained in another. This also includes lexical information — a system may contain a separate concept dictionary, part-of-speech dictionary and semantic lexicon, for example (Fisher *et al.* 1995). On-line tools (*e.g.*, WordNet) can be used to extend these knowledge bases, and the user may be involved in disambiguating them (*e.g.*, when the same word has multiple senses).

Who develops what stages will also depend on the degree to which the stages are domain-dependent. For example in FASTUS, system developers construct the domain-independent modules (*e.g.*, the part-of-speech tagger is trained *a priori* on a large corpus), while domain-dependent modules are left to the end-user to develop (Appelt *et al.* 1995). Macros are provided to support the development of these domain-dependent modules.

The separation of knowledge by stages is important to the design of the system, justified functionally (since each stage relies on the output of earlier stages, it is difficult to develop them out of sequence) and on engineering principles (it is easier to maintain a system in which stages can be modified independently). However, this may not be the optimal organization for presenting information to the user, a point we will return to in the section describing our proposed system.

Most IE systems need to be tailored to a specific corpus of text (an ambitious exception being LOLITA, intended as a general-purpose natural language understanding system that can perform information extraction as one of its tasks (Morgan *et al.* 1995)). Tailoring often begins by tagging the corpus with syntactic and semantic information, either automatically by the system on the basis of existing information (*e.g.*, part-of-speech tagging often relies only on rules derived *a priori* from a large non-domain-specific corpus), or manually by the end-user. In systems such as HASTEN (Krupka 1995) and Alembic Workbench (Day *et al.* 1997), the end-user tags the corpus using a graphical interface and this tagging process forms the basis for communication between user and system — the system derives rules from these tags, applies the rules to tag new text, which the user may either accept or modify, and the cycle repeats.

In general terms, this corpus-based cycle occurs in all systems:

- the system learns patterns with which words occur in the corpus (and possibly other syntactic and semantic information with which the system has augmented the text) and represents them as rules;
- the system applies these rules to a segment of text and the user examines the results;

- the user accepts or modifies the rules.

However, systems differ in the details. We have seen that HASTEN and the Alembic Workbench incorporate this cycle into the tagging process. These systems and most others also allow the user to directly examine and modify the system's rule set (which became the motivation for developing an easier-to-use rule language in FASTUS, called FASTSPEC). As well, systems differ in their ability to handle patterns in the corpus that do not match rules in their rule base. For example, the interfaces in HASTEN and the Alembic Workbench support the user defining how current rules can be generalized to accept new patterns, and CRYSTAL can often generalize its rules automatically (Soderland *et al.* 1995).

Mixed-initiative in IE

The advantage of a mixed-initiative approach is that, in theory, it enables the system to take over as much processing as possible within the constraints of a specific situation (*e.g.*, for a specific corpus, user, and information need) while the user actively ensures the accuracy and appropriateness of the system's results. The system's role is supported by its learning capabilities and its knowledge sources — a knowledge base of rules (learned during the task or *a priori*) and lexical information (drawn from the corpus or from on-line tools such as dictionaries or WordNet). The user's role is supported by knowledge of his or her own information needs. As well:

> "...there are two important advantages that a human expert might have over the machine algorithm: linguistic intuition and world knowledge. Rules that include references to a single lexeme can be expanded to more general applicability by the human expert who is able to predict alternatives that lie outside the current corpus available to the machine. By supporting multiple ways in which rules can be hypothesized, refined and tested, the strengths of both sources of knowledge can be brought to bear." (Day *et al.* 1997)

This introduces the issue of finding the right balance between the roles of user and system. As there is no way of determining this balance ahead of time, negotiation between user and system is necessary.

The Alembic Workbench takes an interesting and effective approach and offers important insights about integrating mixed initiative into information extraction. The user and system take turns annotating text, examining and possibly modifying each other's annotations. Different "stopping criteria" can be defined for indicating when the system's turn in processing text should end, and these may be fixed (*e.g.*, after a certain number of rules) or based on system performance (*e.g.*, when performance improvement falls below some threshold).

An element of mixed initiative that is missing from the Alembic Workbench, however, is the negotiation between system and user — in the Alembic Workbench, criteria are set by the user. Alternatively, a system could give feedback to the user about its own performance and to suggest how the user's information needs and other constraints might be achieved more effectively.

Proposed system

We have discussed current approaches to the task of information extraction and argued that they do not offer sufficient support to a casual user, but have offered no alternatives. In this section, therefore, we introduce a representation for the system knowledge and describe the role that learning can play in a new mixed-initiative approach. At present, we are investigating this approach using a corpus of articles from the Ottawa Citizen newspaper (made available through the Oxford English Dictionary project at the University of Waterloo).

Representing the domain of inquiry

Concepts in the system knowledge base are presented to the user in three separate but interconnected models, providing three "windows" through which to visualize the corpus:

- **query model** — formalized concepts referred to in the user request, defined primarily by the user;
- **domain model** — concepts relevant to the query domain supporting the identification of query model concepts, defined co-operatively by the user and the system;
- **corpus model** — concepts not specific to the domain model, defined primarily by the system but available to the user.

The reason for separating concepts into these three models is that each is understood differently by the user, and acquired and processed differently by the system. Within-domain learning, for example, emphasizes the domain model, while learning across domains mainly refers to the corpus model. This also provides a more coherent context for visualizing the represented knowledge — the query model for visualizing query-related information, the domain model for domain-related information.

Returning to the layoff query described earlier, its elements (`Layoff-Company`, `Layoff-EmployeesAffected`, and `Layoff-StartDate`) are defined in the query model. Next, although the query does not ask about the reason for the layoff, the system may nevertheless need to be able to recognize one if it is given (*e.g.*, to distinguish between forced and voluntary retirement, the first being a form of layoff but not the second). Therefore, the concept of `PlantClosure` appears in the domain model, as would other concepts relevant to layoffs such as `Company` and `Employee`. Finally, the corpus model contains such concepts as `City` and `People` that are not specific to the domain. A query concept such as Layoff-EmployeesAffected is a specialization of, and therefore defined in terms of, the domain concept Employee, which in turn is defined in terms of the corpus concept People.

On-line tools could be used to extend concept definitions, as well. However, some user involvement is necessary to repair information that is missing from these tools, or that is obsolete or inappropriate to the domain of inquiry. For example, WordNet 1.5 returns the following hyponyms of the verb "lay-off":

```
discontinue, stop, cease, give up,
quit, lay off
        => drop, knock off
        => retire, withdraw
        => shut off, close off
        => cheese
```

With only this information, it would be difficult to distinguish a forced retirement (a kind of layoff) from a voluntary one. The assistance of on-line dictionaries is equally problematic. The on-line Webster's (version 7), for example, gives 25 different senses of "layoff" as a verb that the IE system would have to choose from somehow, and the on-line Oxford English Dictionary (2nd Edition) returns no definition but only full-sentence quotations from the years 1904-1952, none of which are relevant here.

We have already described that current systems separate rules in their knowledge base by the different stages at which they are applied. These include rules for recognizing concepts described in the text, for processing sentence structure (syntactic and semantic), for relating coreferent concepts to one another, and for linking concepts to elements in the query template. The casual user, however, will understand these rules better in the context of the text and, we believe, in the context of a specific model. The external representation of the knowledge base, therefore, takes this into consideration.

The role of learning

A number of IE systems are able to learn rules and lexical information automatically from text (*e.g.*, (Day *et al.* 1997) (Weischedel 1995) (Fisher *et al.* 1995)), thereby reducing the need for the user to define them. As well, rules can be generalized to apply to novel sentence constructs — two similar rules can be merged into a single more general rule. The Alembic Workbench (Day *et al.* 1997) supports a number of ways to acquire new rules: they can be hand-crafted (*e.g.*, finite-state phraser rules) or modified directly by the user, learned automatically from annotations in the text (*e.g.*, part-of-speech rules), or learned automatically by inference from other rules (*e.g.*, by generalizing existing rules and rating the performance of the new rules). Both the Alembic Workbench and HASTEN (Krupka 1995) provide graphical support for the user refining or generalizing the system's rules. Modules for the University of Massachusetts system learn and generalize concept definitions, and the relation of concepts to elements of the query template (Fisher *et al.* 1995).

While incorporating learning into an IE system has the potential advantage of reducing the amount of work required of the user, making faulty inferences from the text and thus learning incorrect rules or concepts would result in more work rather than less. Therefore, the user must supervise the system's learning process. An important question, then, is how often must this supervision take place? This issue is not often discussed in IE literature, but it appears that the user is expected to supervise the learning one stage at a time, as each stage builds on the knowledge of earlier stages. It is unclear how large the increments of text should be before the system provides the user with information about what it has learned so far.

A challenging feature of our proposed system is to support a "true" mixed-initiative approach, whereby the decision of what learning strategy to use is negotiated between the system and the user.

At one extreme, a system could adopt a strategy of learning over large increments of text and focusing on one construct at a time. For example, a system may learn the semantic case frame of the verb "layoff" by finding every occurrence of the word in the corpus, and deriving rules for recognizing its arguments. We will refer to such a strategy as "iterative", because it involves iterating over a large amount of text for each construct. The opposite extreme, an "incremental" strategy, involves focusing on a smaller increment of text (*e.g.*, a single document) and deriving rules for recognizing a broader set of constructs. For example, a user unfamiliar with layoffs may not be aware that it is important to process the reason for a job loss (*e.g.*, to distinguish between voluntary and forced retirement); with an incremental approach, the user would be more likely to notice that a layoff was not recognized, and would add a Layoff-Reason concept into the domain model before advancing to the next document. Thus an iterative strategy has the advantage that the learning algorithm is provided with more training data, while an incremental strategy provides more opportunity for the user to correct the system early.

The Alembic Workbench provides the user several possible criteria with which to determine the increments of text used for learning (*e.g.*, a fixed measure such as the number of documents examined, or a variable measure such as the rate of performance improvement). However, it is not clear to what extent the system provides information for the user to decide which criteria are most appropriate at any point during the process of developing the system's knowledge base. It is also not clear whether the user can change the criteria dynamically during the course of an IE task, for example to interrupt the system if too large a learning increment had been selected. We intend to investigate these details.

One possible effect of our proposed approach is that the system attempts to learn on the basis of small increments of text. The advantage of this approach is that the representation of the knowledge base presented to the user is updated frequently, allowing the user to make informed decisions about when to interrupt the automatic learning process. The possible disadvantage, of course, is that the text increments are too small to provide the system with sufficient information for learning. A solution to this dilemma may be to allow learning to be interleaved across stages in the system.

Conclusions

We have demonstrated a starting point for the goal of designing an information extraction system designed to support the casual user, who may be neither an expert in linguistics nor in the domain of inquiry. This approach combines a knowledge representation scheme that divides information according to its relevance to the query domain, and learning techniques for extending and refining this information, into a mixed-initiative strategy. The intended benefits of this approach are to reduce the time and effort required to adapt a system to a new domain of inquiry, and ultimately to increase the accuracy with which information is extracted from text.

A mixed-initiative approach raises several new questions for information extraction, however. For example, what criteria are important for deciding when the system should update information presented to the user (*e.g.*, newly learned concepts and rules)? What performance measures are appropriate for approaches that rely heavily on the interaction between system and user? The traditional measures do not take into consideration the fact that, given sufficient time and user-involvement, a system might achieve perfect performance for a given corpus, nor do they take into consideration the amount of work required to do so. We leave these questions for future investigation.

References

AAAI'97. 1997. *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 97) and the 9th Conference on Innovative Applications of Artificial Intelligence (IAAI 97)*, Providence, Rhode Island: AAAI Press / MIT Press: Menlo Park, CA.

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.

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.: ACL.

Fisher, D.; Soderland, S.; McCarthy, J.; Feng, F.; and Lehnert, W. 1995. Description of the UMass system as used for MUC-6. In MUC-6 (1995), 127–140.

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 AAAI'97 (1997), 992–999.

Holowczak, R. D., and Adam, N. R. 1997. Information extraction based multiple-category document classification for the global legal information network. In AAAI'97 (1997), 1013–1018.

Krupka, G. R. 1995. SRA: Description of the SRA system as used for MUC-6. In MUC-6 (1995), 221–235.

Morgan, R.; Garigliano, R.; Callaghan, P.; Poria, S.; Smith, M.; Urbanowicz, A.; Collingham, R.; Costantino, M.; Cooper, C.; and the LOLITA Group. 1995. University of Durham: Description of the LOLITA system as used in MUC-6. In MUC-6 (1995), 71–85.

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

Okurowski, M. E. 1993. Information extraction overview. In TIPSTER-I (1993), 117–121.

Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95).* Montréal, Québec, Canada: Morgan Kaufman Publishers, Inc. 1314–1319.

TIPSTER-I. 1993. *TIPSTER Text Program Phase I: Workshop Proceedings*, Fredricksburg, Virginia: Morgan Kaufmann: San Francisco, USA.

Weischedel, R. 1995. BBN: Description of the PLUM system as used for MUC-6. In MUC-6 (1995), 55–69.

Will, C. A. 1993. Comparing human and machine performance for natural language information extraction: Results for the TIPSTER text evaluation. In TIPSTER-I (1993), 179–193.