Challenges in Information Extraction from Text for Knowledge Management

Fabio Ciravegna, *University of Sheffield*

Nowadays, most knowledge is stored in an unstructured textual format. We can't query it in simple ways, thus automatic systems can't use the contained knowledge and humans can't easily manage it. The traditional knowledge management process for knowledge engineers has been to manually identify and extract knowledge—a complex and time-consuming process that requires a great deal of manual input. As an example consider the collection of interviews to experts (*protocols*) and their analysis by knowledge engineers in order to codify, model and extract the knowledge of an expert in a particular domain. In this context, *information extraction* from texts is one of the most promising areas of human language technology for KM applications.

Information extraction

IE is an automatic method for locating important facts in electronic documents—for example, information highlighting for enriching a document or storing information for further use (such as populating an ontology with instances). IE thus offers the perfect support for knowledge identification and extraction, because it can, for example, provide support in protocol analysis either in an automatic (unsupervised extraction of information) or semiautomatic way (helping knowledge engineers locate the important facts in protocols through information highlighting).

It is widely agreed that the main barrier to using IE is the difficulty in adapting IE systems to new scenarios and tasks. Most of the current technology still requires the intervention of IE experts. This makes IE difficult to apply, because personnel skilled in IE are difficult to find in industry, especially in small-to-medium-size enterprises.[7] A main challenge is to enable personnel with knowledge on AI (for example, knowledge engineers) who have no or scarce preparation in IE and computational linguistics to build new applications and cover new domains. This is particularly important for KM. IE is just one of the many technologies for building complex applications: wider acceptance of IE will come only when IE tools don't require any specific skill apart from notions of KM.

Several machine-learning-based tools and methodologies are emerging,[8,9] but the road to fully adaptable and effective IE systems is still long. Here, I focus on two main challenges for IE adaptivity in KM that are paramount in the current scenario: automatic adaptation to different text types and human-centered issues in coping with real users.

Adaptivity to Text Types

Porting IE systems means coping with four (often overlapping) main tasks:

1. *Adapting to the new domain information*—implementing system resources such as lexica, knowledge bases, and so forth, and designing new templates so that the system can manipulate domain-specific concepts.
2. *Adapting to different sublanguages features*—modifying grammars and lexica to enable the system to cope with specific linguistic constructions that are typical of the application or domain.
3. *Adapting to different text genres*—specific text genres such as medical abstracts, scientific papers, and police reports might have their own lexis, grammar, and discourse structure.
4. *Adapting to different types*—Web-based documents can radically differ from newspaper-like texts. We need to be able to adapt to different situations.

Technical report no XXXXX also published as
IEEE Intelligent Systems and Their Applications, November 2001, (Trend and Controversies)
This is the tech report version!!!!

Most of the literature on IE has focused on issues 1, 2, and 3, with limited attention to text types, focusing mainly on free newspaper-like texts.[10] This is a serious limitation for portability, especially for KM, where an increase in the use of Inter- and intranet technologies has moved the focus from free-texts-only scenarios (based on, for example, reports and protocols) to more composite scenarios including semi- and structured texts such as highly structured Web pages produced by databases. In classical natural language processing (NLP), adapting to new text types is generally considered a task of porting across different types of free texts.

Using IE for KM requires extending the concept of text types to new, unexplored dimensions. Linguistically based methodologies used for free texts can be difficult to apply or even ineffective on highly structured texts, such as the Web pages databases produce. They can't cope with the variety of extralinguistic structures such as HTML tags, document formatting, and stereotypical language that convey information in such documents. On the other hand, wrapper-like algorithms designed for highly structured HTML pages are largely ineffective on unstructured texts (for example, free texts). This is because such methodologies make scarce or no use of NLP, usually avoiding any generalization over the flat word sequence and tending to be ineffective on free texts, because of, for example, data sparseness.[11]

The challenge is developing methodologies that can fill the gap between the two approaches and cope with different text types. This is particularly important for KM with its composite Web-based scenarios, because Web pages can contain documents of any type and even a mix of text types—an HTML page, for instance, can contain both free and structured texts. Work on this topic has just started.

Wrapper induction systems based on lazy NLP[11] try to learn the best and most reliable level of language analysis useful for a specific IE task by mixing deep linguistic and shallow strategies. The learner starts inducing rules that make no use of linguistic information, such as in wrapper-like systems. It then progressively adds linguistic information to its rules, stopping when the use of NLP information becomes unreliable or ineffective. Generic NLP modules and resources provide linguistic information that is defined once and is not to be modified to specific application needs by users. Pragmatically, the measure of reliability here is not linguistic correctness (immeasurable by incompetent users) but effectiveness in extracting information using linguistic information as opposed to using shallower approaches.

Unlike previous approaches in which different algorithm versions with different linguistic competence were tested in parallel and the most effective version was chosen,[12] lazy NLP-based learners learn which is the best strategy for each information or context separately. For example, they might decide that parsing is the best strategy for recognizing the speaker in a specific application on seminar announcements but not the best strategy to spot the seminar location or starting time. This is promising for analyzing documents with mixed genres.

Coping with non-IE experts

The second main task in adaptive IE concerns human–computer interaction during application development. Nonexpert users must be supported during the entire adaptation process to maximize the final application's effectiveness and appropriateness. A typical IE application's life cycle is composed of scenario design, system adaptation and results validation, and application delivery.[13]

Scenario design

Scenario design defines the information to extract. Many potential users need specific support, because they might find it difficult to manipulate IE-related concepts such as templates. Moreover, there might be a gap between the information the user needs, the information the texts contain, and what the system can actually extract. It is thus important to help users recognize such discrepancies, forcing them into the right paradigm of scenario design. Highlighting information in different colors is generally a good approach. Tag-based interfaces, such as Mitre's Alembic, have proven to be effective and have become a standard in adaptive IE.

Selecting the corpus to be tagged for training is also a delicate issue. Nonlinguistically aware users tend to focus on text content rather than on linguistic variety. Unfortunately, IE systems learn from both. Provided corpora might be unbalanced with respect to types or genres (emails could be underrepresented with respect to free texts) or might show peculiar regularities due to wrong selection criteria. For example, in designing an application on IE from professional resumes, our user selected the corpus by using the names of US cities as keywords. When the trained system was tested , it became clear that most of the resumes actually originated from Europe, where addresses, titles of study, and even text style can significantly differ from US styles. The resulting system was therefore largely ineffective and left the user dissatisfied with the final application.

A number of methodologies can be used to validate the training corpus with respect to—a (hopefully big) untagged corpus. One possible validation concerns the formal comparison of training and untagged corpus. Adam Kilgarriff proposes heuristics for discovering differences in text types among corpora.[14] Average text length, distributions of HTML tags and hyperlinks in Web pages, average frequency of lexical classes in texts (such as nouns), and so forth can be relevant indicators of corpus representativeness and can warn inexperienced users that some training corpora might not sufficiently represent the whole corpus. Even detecting an excess of regularity in the training corpus can indicate an unbalanced corpus selection. For example, if a high percentage of fillers for some slots is the same string (e.g. "ACME Inc.") there is the concrete risk that the corpus contains some unwanted regularity that could influence the learner in an unpredictable way

System adaptation and Results validation

With a corpus reasonable in size and quality, the IE system can then be trained. Unfortunately, even the best algorithm is unlikely to provide optimized results for specific use. This is because a 100 percent accurate system is out of reach for current IE technology, and therefore we must balance recall (the ability to retrieve information when present) and precision (the ratio of correct information on the total of information extracted) to produce the optimal results for the task and users at hand. Different uses will require different types of results—higher recall in some cases, higher precision in others. Users must be enabled to evaluate results from both a quantitative and qualitative point of view and to change the system behavior if necessary. Most of the current technology provides satisfying tools for results inspection: the MUC scorer let users understand the system effectiveness in details.[15] The challenging step now is to let users change system behavior. In case of occasional or inexperienced users, the issue of avoiding technical or numerical concepts such as precision and recall arises. This requires the IE system to bridge the user's qualitative vision ("you are not capturing enough information") with the numerical concepts the learner can manipulate—for example, moving error thresholds to obtain higher recall.

Application delivery

When the application is tuned to specific user needs, it can be delivered and used in the application environment. Corpus monitoring should be enabled even after delivery, though. One of the risks in highly changing environments such as the Internet is that information such as Web pages can change format in a short period of time, and the system must be able to detect such changes.[16] The same techniques mentioned earlier for testing corpus representativeness can identify changes in the information structure or test type.

Adaptive IE is already providing useful results and technology for KM. Fully integrated user-driven solutions are still to come, but current research results are promising.

Acknowledgments

The author's work on adaptive IE is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01). AKT comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton, and the Open University (www.aktors.org).

References

7. F. Ciravegna, A. Lavelli, and G. Satta, "Bringing Information Extraction Out of the Labs: The Pinocchio

Environment," *Proc. 14th European Conf. Artificial Intelligence*, IOS Press, **Berlin,** 2000. **416-421**

8. D. Freitag, "Information Extraction from HTML: Application of a General Learning Approach," *Proc. 15th Nat'l Conf. Artificial Intelligence* (AAAI-98), AAAI Press, Menlo Park, 1998. **517-523**
9. F. Ciravegna, "Adaptive Information Extraction from Text by Rule Induction and Generalisation," *Proc. 17th Int'l Joint Conf. Artificial Intelligence* (IJCAI 2000), **Morgan Kaufmann Publisher, Seattle,** 2001. **1251-1256**
10. C. Cardie, "Empirical Methods in Information Extraction," *AI Journal*, vol. 18, no. 4, **Winter** 1997, pp. 65–79.
11. F. Ciravegna: "(LP)2, An Adaptive Algorithm for Information Extraction from Web-Related Texts," *Proc. IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, **AAAI Working Notes,** 2001. **9-17**
12. S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," *Machine Learning*, Machine Learning 34 (1), February 1999, 233--272.
13. F. Ciravegna and D. Petrelli, "User Involvement in Adaptive Information Extraction: Position Paper" *Proc. IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, **AAAI Working Notes, Seattle,** 2001. **71-73**
14. A. Kilgarriff, "Comparing Corpora,""Comparing Corpora" To appear in International Journal of Corpus Linguistics 15. A. Douthat, *The Message Understanding Conference Scoring Software User's Manual*, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_sw/muc_sw_manual.html
16. I. Muslea, S. Minton, and C. Knoblock, "Co-testing: Selective Sampling with Redundant Views," *Proc. 17th Nat'l Con. Artificial Intelligence* (AAAI-2000) AAAI Press, Menlo Park, Austin, Tx, 2000

Fabio Ciravegna is a senior research fellow in the Department of Computer Science at the University of Sheffield, UK. His research interests include classical and adaptive information extraction from text, with a particular focus on user-centered methodologies. His work on IE has been published at a number of international conferences such as EACL-1999, IJCAI-1999, ECAI-2000, and IJCAI2001. He coorganized two workshops on the use of Machine Learning for Information Extraction at ECAI-2000 and IJCAI2001. Contact him at f.ciravegna@dcs.shef.ac.uk.

Technical report no XXXXX also published as
IEEE Intelligent Systems and Their Applications, November 2001, (Trend and Controversies)
This is the tech report version!!!!