Using HLT for Acquiring, Retrieving and Publishing Knowledge in AKT: Position Paper

K. Bontcheva, C. Brewster, F. Ciravegna, H. Cunningham, L. Guthrie, R. Gaizauskas, Y. Wilks
Department of Computer Science, the University of Sheffield,
Regent Court, 211 Portobello Street, S1 4DP Sheffield, UK

Email: N.Surname@dcs.shef.ac.uk

Abstract

AKT is a major research project applying a variety of technologies to knowledge management. Knowledge is a dynamic, ubiquitous resource, which is to be found equally in an expert's head, under terabytes of data, or explicitly stated in manuals. AKT will extend knowledge management technologies to exploit the potential of the semantic web, covering the use of knowledge over its entire lifecycle, from acquisition to maintenance and deletion. In this paper we discuss how HLT will be used in AKT and how the use of HLT will affect different areas of KM, such as knowledge acquisition, retrieval and publishing.

1 Introduction

As globalisation reduces the competitive advantage existing between companies, the role of proprietary information and its appropriate management becomes all-important. A company's value depends more and more on "intangible assets"[1] which exist in the minds of employees, in databases, in files and in a multitude of documents. It is the goal of knowledge management (KM) technologies to make computer systems which provide access to this intangible knowledge present in a company or organisation. The system must make it possible to share, store and retrieve the collective expertise of all the people in an organization. At present, many companies spend considerable resources on knowledge management; estimates range between 7 and 10% of revenues (Davenport 1998).

In developing a knowledge management system, the knowledge must first be captured or acquired in some form which is usable by a computer. The knowledge acquisition bottleneck, so well-known in AI, is just as important in knowledge management. The acquisition of knowledge does not become less difficult in a business environment and often requires a sea-change in company culture in order to persuade users to accommodate to the technology adopted, precisely because knowledge acquisition is so difficult.

Once knowledge has been acquired, it must be managed, i.e. modelled, updated and published. Modelling means representing information in a way that is both manageable and easy to integrate with the rest of the company's knowledge. Updating is necessary because knowledge is dynamic. Part of its importance for a company or individual lies in the fact that knowledge is ever changing and keeping up with the change is a crucial dimension in knowledge management. Publishing is the process that allows sharing the knowledge across the company. These needs have crystallised in efforts to develop the so-called Semantic Web. It is envisaged that in the future, the content currently available on the Web (both Internets and Intranets) as raw data will be automatically annotated with machine-readable semantic information. In such a case, we will no longer speak of information retrieval but rather of Knowledge Retrieval because instead of obtaining thousands of potentially relevant or irrelevant documents, only the dozen or so documents that are truly needed by the user will be presented to them.

[1] A term coined by Karl-Erik Sveiby

In this paper we present the way Human Language Technology (HLT) is used to address several facets of the KM problem: acquiring, retrieving, and publishing knowledge. The work presented in this paper is supported by the AKT project (Advanced Knowledge Technologies), a multimillion pound six year research project funded by the EPSRC in the UK. AKT, started in 2000, involves the University of Southampton, the Open University, the University of Edinburgh, the University of Aberdeen, and the University of Sheffield together with a large number of major UK companies. Its objectives are to develop technologies to cope with the six main challenges of knowledge management:

- acquisition
- modelling
- retrieval/extraction
- reuse
- publication
- maintenance

These challenges will be addressed by the University of Sheffield in the context of AKT by the application of a variety of human language technologies. Here, we consider only the contribution of HLT to the acquisition of knowledge, its retrieval and extraction, its publication, and finally the role of appropriate HLT infrastructure to the completion of these goals.

2 Knowledge Acquisition

Knowledge acquisition (KA) is concerned with the process of turning data into coherent knowledge for a computer program. The need for effective KA methods increases as the quantity of data available electronically increases year by year, and the importance it plays in our society is more and more recognised. The challenge, we believe, lies in designing effective techniques for acquiring the vast amounts of (largely) tacit knowledge. KA is a complex process, which traditionally is extremely time consuming.

Existing KA methodologies are varied but almost always require a great deal of manual input. One methodology, often used in Expert Systems, involves the time-consuming process of structured interviews ('protocols'), which are then analysed by knowledge engineers in order to codify and model the knowledge of an expert in a particular domain. Even if a complex expert system is not required, all forms of KA are very labour intensive. Yahoo currently employs over 100 people to keep its category hierarchy up to date (Dom 1999). Some methodologies have started to appear to automate this process, although still limited to some steps in the KA process. They depend on replacing the introspection of knowledge engineers or the extended elicitations of the protocol methods (Ericsson and Simon 1984) by using Human Language Technologies, more specifically Information Extraction, Natural Language Processing and Information Retrieval.

Although knowledge acquisition produces data (knowledge) for use by a computer program, the form and content of that knowledge is often debated in the research community. Ontologies have emerged as one of the most popular means of modelling the knowledge of a domain. The meaning of this word varies somewhat in the literature, but minimally it is a hierarchical taxonomy of categories, concepts or words. Ontologies can act as an index to the memory of an organisation and facilitate semantic searches and the retrieval of knowledge from the corporate memory as it is embodied in documents and other archives. Repeated research has shown their usefulness, especially for specific domains (Järvelin and Kekäläinen 2000). The process of ontology construction is illustrated in the rest of this section.

2.1 Taxonomy construction

We propose to introduce automation in the stage of taxonomy construction mainly in order to eliminate or reduce the need for extensive elicitation of data. In the literature approaches to construction of taxonomies of concepts have been proposed (Brown *et al.* 1992, McMahon and Smith 1996, Sanderson and Croft 1999). Such approaches either use a large collection of documents as their sole data source, or they can attempt to use existing concepts to extend the taxonomy (Agirre *et al.*2000, Scott 1998). We intend to develop a semi-automatic method that, starting from a seed ontology sketched by the user, produces the final ontology via a cycle of refinements by eliciting knowledge from a collection of texts. In this approach the role of the user should only be that of proposing an

initial ontology and validating/changing the different versions proposed by the system.

We intend to integrate a methodology for automatic hierarchy definition (such as (Sanderson and Croft 1999)) with a method for the identification of terms related to a concept in a hierarchy (such as (Scott 1998)). The advantage of this integration is that, as knowledge is continually changing, we can reconstruct an appropriate domain specific ontology very rapidly. This does not preclude incorporating an existing ontology and using the tools to extend and update it on the basis of appropriate texts. Finally an ontology defined in this way has the particular advantage that it overcomes the well-known 'Tennis problem' associated with many predefined ontologies such as WordNet, i.e where terms closely related in a given domain are structurally very distant such as *ball* and *court*, for example.

In addition we intend to employ classic Information Extraction techniques (described below) such as named entity recognition (Humphreys *et al.* 1998) in order to pre-process the text, as the identification of complex terms such as proper names, dates, numbers, etc, allows to reduce data sparseness in learning (Ciravegna 2000).

We plan to introduce many cycles of ontology learning and validation. At each stage the defined ontology can be: i) validated/corrected by a user/expert; ii) used to retrieve a larger set of appropriate documents to be used for further refinement (Järvelin and Kekäläinen 2000); iii) passed on to the next development stage.

2.2 Learning Other Relations

This stage proceeds to build on the skeletal ontology in order to specify, as much as possible without human intervention, relations among concepts in the ontology, other than ISAs. In order to flesh the concept relations, we need to identify relations such as synonymy, meronymy, antonymy and other relations. We plan to integrate a variety of methods existing in the literature, e.g. by using recurrences in verb subcategorisation as a symptom of general relations (Basili *et al.* 1998), by using Morin's user-guided approach to identify the correct lexico/syntactic environment (Morin 1999), and by using methods such as (Hays 1997) to locate specific cases of synonymy.

3 Knowledge Extraction

Assuming that the shape of knowledge has been acquired and adequately modelled, it will have to be stored in a repository from which it is retrieved as and when needed. On the one hand there is the problem of retrieving instances in order to populate the resulting knowledge base. On the other hand, considering that repositories could become very substantial in size, there is the necessity to navigate the repository in order to extract the knowledge when needed. In this section we focus on the problem of knowledge base population, as it is in our opinion the most challenging from the HLT point of view.

3.1 Knowledge Base Population

Instance identification for Knowledge Base population can be performed by HLT-based document analysis. With the term documents, we mean a wide variety of types of texts such as plain texts, web pages, knowledge elicitation interview transcriptions (protocols), etc. For the sake of this paper we limit our analysis to language related tasks only, ignoring the problem of multi-media information. As a first step instance identification requires the identification of relevant documents containing citation of the interesting information (document classification). Then it requires the ability to identify and extract information from documents (Information Extraction from text).

3.2 Document Classification

Text classification for IE purposes has been explored both in the MUC conferences as well as in some commercially oriented projects (Ciravegna *et al.* 2000). In concrete terms classification is used in order to identify the scenario to apply to a specific set of texts, while IE will identify (i.e. index) the instances in the texts. In most cases of application document classification is quite straightforward, being limited to the Boolean classification of a document between relevant/irrelevant (single scenario application as in the MUC conferences). In cases in which knowledge may be distributed along a number of different detailed scenarios, full document classification

is then needed. In such cases, two main characteristics are relevant for the classification approach: flexibility and refinability (Ciravegna *et al.* 1999). **Flexibility** is needed with respect to both the number of the categories and the granularity of the classification to be coped with. Three main types of classification can be identified: coarse-grained, fine-grained, and content-based. *Coarse-grained* classification is performed among a relatively small number of classes (e.g., some dozens) that are sharply different (e.g., sport vs finance). This can be obtained reliably and efficiently by the application of statistical classifiers. *Fine-grained* classification is performed over a usually larger number of classes that can be very similar (e.g., discriminating between news about private bond issues and news about public bond issues). This type of classification generally requires some more knowledge-oriented approaches such as pattern-based classification. Sometimes categories are so similar that classification needs to be *content-based*, i.e. it can be performed only by extracting the news content (e.g., finding news articles issued by English financial institutions referring to amounts in excess of 100,000 Euro). In this case some forms of shallow adaptive Information Extraction can be used (see next section). **Refinability** concerns the possibility of performing classification in a sequence of steps, each one providing a more precise classification (from coarse-grained to content-based). In the current technological situation coarse-grained classification can be performed quickly, while the systems available for more fine-grained classification are much slower and less general purpose. When the amount of textual material is large an incremental approach, based on some level of coarse-grained classification further refined by successive analysis, proves to be very effective. A refinable classification is generally performed over a hierarchy of classes. A refinement may revise the categories assigned to specific texts with more specialised classes from the hierarchy. More complex techniques are invoked only when needed and, in any case, within an already detected context (Ciravegna *et al.* 1999).

We plan to produce a number of solutions for text classification, adaptable to different scenarios and situations, following the criteria mentioned above.

3.3 Information Extraction

Information extraction from text (IE) is the process of mapping of texts into fixed format output (templates) representing the key information (Gaizauskas 1997). In using IE for KM, templates represent an intermediate format for mapping the information in the texts into ontology instances. Templates can be semi-automatically derived from the ontology. We plan to use IE for a number of passes: on the one hand, we plan to populate a knowledge base with instances as mentioned above. On the other hand, IE can be used to monitor relevant changes in the information, providing a fundamental contribution to the problem of knowledge updating. We have a long experience in IE from texts, Sheffield having actively participated in the MUC conferences and in the TIPSTER project, activities that historically have made a fundamental contribution to making IE as we now know it. The new challenge we are currently addressing is adaptivity. Adaptivity is a major goal for Information Extraction, especially in the case of its application to knowledge management, as KM is a process that has to be distributed throughout companies. The real value of IE will become apparent when it can be adapted to new applications and scenarios directly by the final user without the intervention of IE experts. The goal for research in adaptive IE is to create systems adaptable to new applications/domains by using only an analyst's knowledge, i.e. knowledge about the domain/scenario.

There are two directions of research in adaptive IE, both involving the use of Machine Learning. On the one hand machine learning is used to automate as much as possible the tasks an IE expert would perform in application development (Cardie 1997) (Yangarber *et al.* 2000). The goal here is to reduce the porting time to a new application (and hence the cost). This area of research comes mainly from the MUC community. Currently, the technology makes use mainly of NLP-intensive technologies and the type of texts addressed are mainly journal articles.

On the other hand, there is an attempt to make IE systems adaptable to new

domains/applications by using only an analyst's knowledge, i.e. knowledge about the domain/scenario only (Kushmerick *et al.* 1997), (Califf 1998), (Muslea *et al.* 1998), (Freitag and McCallum 1999), (Soderland 1999), (Freitag and Kushmerick 2000), (Ciravegna 2001a). Most research has so far focused on Web-related texts (e.g. web pages, email, etc.) Successful commercial products have been created and there is an increasing interest on IE in the Web-related market. Current adaptive technologies make no use of natural language processing in the web context, as extra linguistic structures (e.g. HTML tags, document formatting, and ungrammatical stereotypical language) are the elements used to identify information. Linguistically intensive approaches are difficult or unnecessary in such cases. When these non-linguistic approaches are used on texts with a reduced (or no) structure, they tend to be ineffective.

There is a technological gap between adaptive IE on free texts and adaptive IE on web-related texts. For the purposes of KM, such a gap has to be bridged so to create a set of technologies able to cover the whole range of potential applications for different kinds of texts, as the type of texts to be analysed for KM may vary dramatically from case to case. We plan to bridge this gap via the use of lazy natural language processing. We intend to use an approach where the system starts with a range of potential methodologies (from shallow to linguistically intensive) and learns from a training corpus which is the most effective approach for the particular case under consideration. A number of factors can influence the choice: from the type of texts to be analysed to the type of information the user is able to provide in adapting the system. In the first case the system will have to identify what type of task is under consideration and select the correct level of analysis (e.g. language based for free texts). Formally in this case the level of language analysis is one of the parameters the learner will have to learn. Concerning the type of tagging the user is able to provide: different users are able to provide different levels of information in training the system: IE-trained users are able to provide sophisticated tagging, maybe inclusive of syntactic, semantic or pragmatic information. Naïve users on the other hand are only able to provide some basic information (e.g. to spot the relevant information in the texts and highlight it in different colours). We plan to develop a system able to cope with a wide of variety of situations by starting from the **(LP)²** algorithm and enhancing its learning capabilities on free texts (Ciravegna 2001) and developing a powerful human-computer interface for system adaptation (Ciravegna and Petrelli 2001).

4 Knowledge Publishing

Knowledge is only effective if it is delivered in the right form, at the right place, to the right person at the right time. Knowledge publishing is the process that allows getting knowledge to the people who need it in a form that they can use. As a matter of fact, different users need to see knowledge presented and visualised in quite different ways. The dynamic construction of appropriate perspectives is a challenge which, in AKT, we will address from the perspective of generating automatically such presentations from the ontologies acquired by the KA and KE methods, discussed in the previous sections. Natural Language Generation (NLG) systems automatically produce language output (ranging from a single sentence to an entire document) from computer-accessible data, usually encoded in a knowledge or data base (Reiter 2000). NLG techniques have already been used successfully in a number of application domains, the most relevant of which is automatic production of technical documentation (Reiter *et al.* 1995), (Paris *et al.* 1996). In the context of KM and knowledge publishing in particular, NLG is needed for knowledge diffusion and documenting ontologies. The first task is concerned with personalised presentation of knowledge, in the form needed by each specific user and tailored to the correct language type and the correct level of details. The latter is a very important issue, because as discussed earlier, knowledge is dynamic and needs to be updated frequently. Consequently, the accompanying documentation which is vital for the understanding and successful use of the acquired knowledge, needs to be updated in sync. The use of NLG simplifies the ontology maintenance and update tasks, so that the knowledge engineer can concentrate on the

knowledge itself, because the documentation is automatically updated as the ontology changes.
The NLG-based knowledge publishing tools will also utilise the ontology instances extracted from documents using the IE approaches discussed in Section 3.3. The dynamically generated documentation will not only include these instances, as soon as they get extracted, but it will also provide examples of their occurrence in the documents, thus facilitating users' understanding and use of the ontology.
Our approach to knowledge publishing is based on an existing framework for generation of user-adapted hypertext explanations (Bontcheva 2001), (Bontcheva and Wilks 2001). The framework incorporates a powerful agent modelling module, which is used to tailor the explanations to the user's knowledge, task, and preferences. We are now also extending the personalisation techniques to account for user interests. The main challenge for NLG will be to develop robust and efficient techniques for knowledge publishing which can operate on large-scale knowledge resources and support the personalised presentation of diverse information, such as speech, video, text, graphics (see (Maybury 2001)).
The other challenge in using NLG for knowledge publishing is to develop tools and techniques that will enable knowledge engineers, instead of linguists, to create and customise the linguistic resources (e.g., domain lexicon) at the same time as they create and edit the ontology. In order to allow such inter-operability with the KA tools, we will integrate the NLG tools in the GATE infrastructure, discussed next.

5 HLT Infrastructure

The range and complexity of the task of knowledge management make imperative the need for standardisation. While there has been much talk about the re-use of knowledge components such ontologies, much less has been undertaken to standardise the infrastructure for tools and their development. The types of data structures typically involved are large and complex, and without good tools to manage and allow succinct viewing of the data we will continue to work below our potential. The University of Sheffield has pioneered in the Gate and Gate 2 projects the development of an architecture for text engineering (Cunningham *et al.* 1997), (Cunningham *et al.* 2000). Given the modular architecture and component structure of Gate, it is natural to build on this basis to extend the capabilities of Gate so as to provide the most suitable possible environment for tool development, implementation and evaluation in AKT. The system will provide a single interaction and deployment point for the roll-out of HLT in Knowledge Management. We expect Gate2 to act as the skeleton for a large range of knowledge management activities within AKT and plan to extend its capabilities within the life of the AKT project by integrating with suitable ontological and lexical databases in order to permit the use of the Gate system with large bodies of heterogeneous data

6 Conclusion and Future Work

We have presented how we plan to use HLT for helping KM in AKT. We believe that HLT can make a substantial contribution to the following issues in KM:

- Cost reduction: KM is an expensive task, especially in the acquisition phase. HLT can aid in automating both the acquisition of the structure of the ontology to be learnt and in populating such ontology with instances. It will also provide support for automatic knowledge documentation.
- Time reduction: KM is a slow task: HLT can help in making it more efficient by reducing the need for the human effort;
- Subjectivity reduction: this is a main problem in knowledge identification and selection. Subjective knowledge is difficult to integrate with the rest of the company's knowledge and its use is somehow difficult.

KM constitutes a challenge for HLT as it provides a number of fields of application and in particular it challenges the integration of a set of techniques for a common goal.

Acknowledgement

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and

Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

References

Agirre, E. O. Ansa, E. Hovy, and D. Martínez 2000. Enriching very large ontologies using the WWW, *Proceedings of the ECAI 2000 workshop "Ontology Learning"*.

Basili, R., R. Catizone, M. Stevenson, P. Velardi, M. Vindigni, and Y. Wilks. 1998. 'An Empirical Approach to Lexical Tuning'. *Proceedings of the Adapting Lexical and Corpus Resources to Sublanguages and Applications Workshop*, held jointly with 1st LREC Granada, Spain.

Bontcheva, K. 2001. Generating adaptive hypertext explainations with a nested agent model. *Ph. D. Thesis*, University of Sheffield.

Bontcheva, K. and Wilks, Y. 2001. Dealing with Dependencies between Content Planning and Surface Realisation in a Pipeline Generation Architecture. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI2001),* Seattle.

Brown, P.F., Peter F., V. J. Della Pietra, P. V. DeSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18, 467-479.

Califf, M. E. 1998. Relational Learning Techniques for Natural Language Information Extraction. *Ph.D. thesis*, Univ. Texas, Austin, www/cs/utexas.edu/users/mecaliff

C. Cardie, `Empirical methods in information extraction', AI Journal,18(4), 65-79, (1997).

F. Ciravegna, A. Lavelli, N. Mana, J. Matiasek, L. Gilardoni, S. Mazza, M. Ferraro, W. J. Black F. Rinaldi, and D. Mowatt. FACILE: Classifying Texts Integrating Pattern Matching and Information Extraction. In *Proceedings of the 16th International Joint Conference On Artificial Intelligence (IJCAI99)*, Stockholm, Sweden, 1999.

F. Ciravegna, A. Lavelli,, L. Gilardoni, S. Mazza, W. J. Black, M. Ferraro, N. Mana, J. Matiasek, F. Rinaldi. Flexible Text Classification for Financial Applications: The *FACILE* System. In *Proceedings of Prestigious Applications sub-conference (PAIS2000) sub-conference of the 14th European Conference On Artificial Intelligence (ECAI2000)*, Berlin, Germany, August, 2000.

Ciravegna, F. 2001. Adaptive Information Extraction from Text by Rule Induction and Generalisation. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI2001),* Seattle.

Ciravegna, F. and D. Petrelli. 2001. User Involvement in customizing Adaptive Information Extraction from Texts: Position Paper. *Proceedings of the IJCAI01 Workshop on Adaptive Text Extraction and Mining,* Seattle.

Cunningham, H., K. Humphreys, R. Gaizauskas and Y. Wilks. 1997. Software Infrastructure for Natural Language Processing. *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97*).

Cunningham H., K. Bontcheva, V. Tablan and Y. Wilks. 2000. Software Infrastructure for Language Resources: a Taxonomy of Previous Work and a Requirements Analysis. *Proceedings of the Second Conference on Language Resources Evaluation*, Athens.

Dom, B. 1999. Automatically finding the best pages on the World Wide Web (CLEVER). *Search Engines and Beyond: Developing efficient knowledge management systems*. Boston, MA.

Ericsson, K. A. and H. A. Simon. 1984. *Protocol Analysis: verbal reports as data*. MIT Press, Cambridge, Mass.

Freitag, D. and A. McCallum. 1999 Information Extraction with HMMs and Shrinkage. *AAAI-99 Workshop on Machine Learning for Information Extraction,* Orlando, FL. (www.isi.edu/~muslea/RISE/ML4IE/)

Freitag, D. and N. Kushmerick. 2000. Boosted wrapper induction. F. Ciravegna, R. Basili, R. Gaizauskas, *ECAI2000 Workshop on Machine Learning for Information Extraction*, Berlin, 2000, (www.dcs.shef.ac.uk/~fabio/ecai-workshop.html)

Hays, P. R. 1997. *Collocational Similarity: Emergent Patterns in Lexical Environments*, PhD. Thesis. School of English, University of Birmingham

Humphreys, K., R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham and Y. Wilks. 1998. Description of the University of Sheffield LaSIE-II System as used for MUC-7.

Proceedings of the 7th Message Understanding Conference.

Järvelin, K. and J. Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, , Athens.

Kushmerick, N., D. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. *Proceedings of 15th International Conference on Artificial Intelligence*, *IJCAI-97*.

Manchester, P. 1999. Survey – Knowledge Management. *Financial Times*, 28.04.99.

Maybury, M.. 2001. Human Language Technologies for Knowledge Management: Challenges and Opportunities. *Workshop on Human Language Technology and Knowledge Management.* Toulouse, France.

McMahon, J. G. and F. J. Smith. 1996 Improving Statistical Language Models Performance with Automatically Generated Word Hierarchies. *Computational Linguistics*, 22(2), 217-247, ACL/MIT.

Morin, E. 1999. Using Lexico-Syntactic patterns to Extract Semantic Relations between Terms from Technical Corpus, *TKE 99*, Innsbruck, Austria.

Muslea, I., S. Minton, and C. Knoblock. 1998. Wrapper induction for semi-structured, web-based information sources. *Proceedings of the Conference on Autonomous Learning and Discovery CONALD-98*.

Paris, C. , K. Vander Linden. 1996. DRAFTER: An interactive support tool for writing multilingual instructions, *IEEE Computer, Special Issue on Interactive NLP*.

Reiter, E. 1995. NLG vs. Templates. Proceedings of the 5th European workshop on natural language generation, (ENLG-95), Leiden.

Reiter, E. , C. Mellish and J. Levine. 1995 Automatic generation of technical documentation. *Journal of Applied Artificial Intelligence*, 9(3) 259-287, 1995

Sanderson, M. and B. Croft. 1999. Deriving concept hierarchies from text. *Proceedings of the 22nd ACM SIGIR Conference*, 206-213.

Scott, M. 1998. Focusing on the Text and Its Key Words. *TALC 98 Proceedings*, Oxford, Humanities Computing Unit, Oxford University.

Soderland, S. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, (1), 1-44.

Yangarber, R., R. Grishman, P. Tapanainen and S. Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. *Proceedings of COLING 2000: The 18th International Conference on Computational Linguistics*, Saarbrücken.