User Centred Ontology Learning for Knowledge Management

Christopher Brewster, Fabio Ciravegna and Yorick Wilks
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
{C.Brewster|F.Ciravegna|Y.Wilks}@dcs.shef.ac.uk

Abstract

Automatic (or semi-automatic) ontology building is a current issue in many application fields where ontologies are currently built manually. This paper presents a user-centred methodology for ontology construction based on the use of Machine Learning and Natural Language Processing. In our approach, the user starts the process by initially sketching a preliminary ontology (or selecting an existing one) and a corpus of relevant texts. Then the learner uses the ontology in order to retrieve examples of lexicalisation of relations (e.g. ISA relation) in the corpus. Retrieved examples are validated by the user and used by the learner to generate patterns to discover other instances of the same relation. New instances added to the existing ontology or used to tune the existing ontology. The discovering process is repeated until a satisfying ontology is obtained. The methodology largely automate the building process. It focuses the expensive user activity on sketching the initial ontology, validating textual examples and the final ontology, while the system performs the tedious and expensive activity of searching a large corpus for knowledge discovery. Moreover the output of the process is not only an ontology, but also a system trained to rebuild and eventually retune the ontology, as the learner is adapted by the user feedback. This simplifies ontology maintenance, a major problem in ontology-based methodologies.

1. Introduction

The importance of ontologies is widely accepted in a number of domains including the Semantic Web, Knowledge Management and electronic commerce (Fensel *et al.* 2001, Berners-Lee *et al.* 2001, Brewster *et al.* 2001). They provide a means to structure and model the concepts share by a group of people concerning a specific domain. They permit a variety of services to be built using them and perhaps most important provide a form of semantics for human-machine interaction. While a great deal of effort is going into the planning of how to use ontologies, much less a has been achieved in automating their construction, in making feasible in effect a computational process of knowledge capture.

The tradition in ontology construction is that this is an entirely manual process. There are large teams of editors or, so-called, 'knowledge managers' who are occupied in editing knowledge bases for eventual use by a wider community in their organisation. Such is the case of the taxonomies built and maintained by as diverse organisations as Yahoo and GlaxoSmithKline. The source of information, of knowledge, for these knowledge structures is usually introspection or, more traditionally, protocol analysis (Ericson and Simon 1984). In this context, the

Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm, June 27-28, 2002, Lecture Notes in Computer Sciences, Springer Verlag.

automation of the process of knowledge capture is still in its infancy. Even if editors such as Protégé (REF) exist, for writing ontologies in a standard format, in essence, they are sophisticated data entry interfaces which do not significantly reduce the labour involved.

The process of knowledge capture or ontology construction involves three major steps: first, the construction of a concept hierarchy; secondly, the labelling of relations between concepts, and thirdly, the association of content with each node in the ontology (Brewster *et al.* 2001). It is clear from the present state of research (e.g. Maedche and Volz 2001) that although much can be gained from using external data sources (existing ontologies, named entity gazeteers, etc.) the continuously changing dynamic nature of human knowledge makes a system that can be trained on real data (texts) an imperative.

In the past a number of researchers have proposed methods for creating conceptual hierarchies or taxonomies of terms by processing texts by applying methods from Information Retrieval (term distribution in documents) and Information Theory (mutual information) (Brewster 2002). It is relatively easy to show that two terms are associated in some manner or to some degree of strength (e.g. Grefenstette 1994, Scott 1998). It is possible also to group terms into hierarchical structures of varying degree of coherence (e.g. Brown *et al.* 1992, Sanderson and Croft 1999). However, the most significant challenge, which has not been resolved, is to be able to label the nature of the relationship between the terms. The importance of this step lies in major part because it acts both as a qualitative evaluation on the effectiveness of a method which merely associates two terms, and as a step towards a more fully specified taxonomy/ontology where the nature of relations are explicit. Only if relations are explicit can an ontology be used with problem solving methods (PSMs) (Gomez-Perez 1999) i.e. for some form of logical inference.[f1]

There is an indefinite number of significant ontological relationships. Some reflect classical linguistic relationships (hyponymy, synonymy), others real world relations (meronymy) and yet others relations specific to a domain ('merger relation', 'firing relation'). Each of these relationships between terms may be reflected in one or more lexico-syntactic patterns in the texts analysed. Thus, for example, the *merger relation* can be expressed in a number of ways:

- *merger of X with Y*
- *X completed the acquisition of Y*
- *X said its shareholders approve the merger with Y*
- *etc.*

Thus for any given ontological relationship it is possible to model such relationship using a corpus of relevant texts and retrieving the relevant lexico-syntactic patterns. Marti Hearst proposes to "identify a set of lexico-syntactic patterns that are easily recognisable, that occur frequently and across text genre boundaries, and that indisputably indicate the lexical relation of interest" (Hearst 1992). Hearst argues that pattern matching is more successful than parsing in identifying patterns in text that reveal hyponymy relations between words. She proposes a number of possible patterns, for example:

such NP as {NP, } 8 {(or| and)} NP e.g.: …works by such authors as Herrick, Goldsmith, and Shakespeare.

NP {, NP} {, } or other NP* e.g.: Bruises, wounds, broken bones or other injuries …

The method by which such patterns are found involves the following steps:

1. Identifying a potential lexical relation of interest e.g. *group/member*
2. Collect a list of exemplars of this relation e.g. *England/country* using MRDs, KBs, etc.
3. Collect a list of citations where these expressions occur syntactically.
4. Identify the commonalities in the syntactic/lexical environment in order to construct a pattern.
5. Use the pattern to collect instances of the target relation.

Hearst did not implement this procedure because she considered step 4 'undetermined'. For one relation (*such as*), out of 8.6M words of encyclopaedia text, she found 7067 sentences which contained the phrase, and of these 152 fitted the pattern. With a slight easing of restrictions, 330 exemplars were found. She does not present enough figures to properly quantify the success of her method which she describes as 'encouraging'. Two challenges arise from Hearst's work. The first is to implement the whole process as much as possible. The second concerns the relatively few exemplars which appear to be found even if one has identified the correct pattern for a specific ontological relationship.

Inspired by Hearst, a first attempt at semi-automating the process has been undertaken by Morin, in a system which in effect is an implementation of the procedure outlined by Hearst in her original paper. In Morin's system, there are seven steps:

1. Select manually a representative conceptual relation, for instance the hypernym relation.
2. Collect a list of pairs of terms linked by this relation. The list of pairs can be extracted from a thesaurus, a knowledge base or can be specified manually. …
3. Find sentences in which conceptually related terms occur. These sentences are lemmatised, and noun phrases are identified. Therefore, sentences are represented as lexico-syntactic expressions. …
4. Find a common environment that generalizes the lexico-syntactic expression extracted at the third step. This environment is calculated with the help of a measure of similarity and a procedure of generalization that produce candidate lexico-syntactic patterns. …
5. Validate candidate lexico-syntactic patterns by an expert.
6. Use new patterns to extract more pairs of candidate terms.
7. validate candidate terms by an expert, and go to step 3.

From Finkelstein-Landau and Morin 1999:2-3

Crucially, Morin has not been able to avoid the intervention of an expert in at least two steps of his process. He is however aware of this limitation and others. He states that it "can find only a small portion of related terms due to the variety of sentence styles and the inability to find a common environment to all those sentences" (Finkelstein-Landau and Morin 1999:6).

The identification and validation of the 'common' lexico-syntactic pattern (or set of patterns) for a given ontological link needs to be designed in such a manner as to maximally exploit the strengths of using a computer and also the strengths of a human expert. Neither in Hearst nor in Morin has there been any attempt to use Machine Learning (ML) methods in identifying the 'common' lexico-syntactic environment. In the following sections, we present our 'co-operative' method, which allows for far greater generality of application and exploits the particular strengths of the machine and the user to achieve a common goal viz. creating an ontology.

2. Building Ontologies for Knowledge Management

We propose an evolution of Hearst and Morin approach to be used to build ontologies for real world applications for KM. Our methodology is based on a co-operative model of user and system interaction. The model is based on the integration of Natural Language Processing techniques (particularly Information Extraction) with user input, so as to limit the user's effort and yet obtain the most accurate possible ontology. Our objective is to make as effective as possible the user's input to the system without expecting any understanding of the nature of (for example) 'lexico-syntactic patterns'. In order to achieve this, we need to have a slightly greater understanding of the qualities of the user and the system.

The Characteristics of the User

The system we are proposing is developed for the specific context and needs of Knowledge Management (KM). An ontology is not being built for or in itself as an intellectual exercise but for the particular needs of KM and this implies a user with specific characteristics. This user is assumed not to have the specialised knowledge of IE, of NLP and of linguistics which might make them able to understand the nature of lexico-syntactic patterns. They would need a considerable understanding of linguistics in order to the various levels of analysis involved (lexical item, lemma, part of speech, syntactic role, etc.). Thus, we assume they are unable to write such patterns themselves. However, the user is assumed able to undertake the following tasks as part of the process of developing an ontology:

1) They are able to draft an ontology, or select or reuse an existing one, and provide this as input to the system.
2) They are able to validate sentences which are exemplars of a particular relation between two terms.

They are able to name/label a relation exemplified in a particular sentence, and to recognise when they encounter further instances of such a relation.

The Characteristics of the System

In a similar manner to the user, we can identify the particular characteristics of the system. To some extent, these are characteristics of computer systems in general, but here we focus specifically on the capabilities of a combined NLP/IE system. In general they are:

1) able to analyse large quantities of texts at speeds which often approximate real time.
2) able to find regularities and identify all occurrences of a given regularity.
3) able to group or cluster words and other patterns into groups.
4) able to easily establish that a relationship exists between any given term *x* and another term *y*.

This ability of computer systems to handle large quantities of data has revolutionised lexicography and should have a similar effect on ontology construction and knowledge capture. The ability to find regularities is particularly significant in view of the large quantities of data involved, and given Zipf's law it can be expected that some regularities are very frequent but that many occur very rarely.

3. User Centred Pattern Learning

We believe that the above characteristics of user and system are complementary and that they must be taken into account in defining a cooperative methodology for ontology building. The goal of our methodology is to produce a (semi-)automatic methodology where user and system interplay in order to maximise the effectiveness of the process while minimizing the user effort.

The learning process is divided in two meta-steps: the system will first attempt to learn about the ISA/hyponymy relations among concepts, as these forms the backbone of all ontologies and often is almost the only relation represented (e.g. the Gene Ontology). Once these have been established (via the steps below) the skeletal ontology is presented to the user and they may select further relations in the input ontology to learn.

Each of the two meta-steps above is organised in three steps: bootstrapping, pattern learning & user validation, and cleanup.

Bootstrapping

The bootstrapping process involves setting up the system – user interaction. The user has to provide the system with some basic data in order to begin the learning process. First, the user must specify an appropriate corpus of texts that the leaner will use. Second, the user must provide a seed ontology. Seed ontologies can vary in size considerably, as they can be large existing ontologies, or can be small user-defined ontologies, e.g. very high-level descriptions of a domain maybe including some more detailed sections specified (e.g. because this is the place where a particularly good range of ontological relations is to be found). The draft ontology must be associated with a small thesaurus of words, i.e. the user must indicate at least one term that lexicalises each concept in the hierarchy.

Pattern Learning & User Validation

Words and relations in the thesaurus are used by the system to retrieve a first set of examples of lexicalisation of relations among concepts in the corpus. The same information can be obtained by asking the user to type an example of lexicalisation for each relation. In this case the system will directly move to the learning step. The sentences identified by the system are then presented to the user for validation. Retrieved examples are to be classified by the user as correct examples (positive examples) or wrong examples of a specific relation (negative examples). The learner then uses positive and negative examples to induce generic patterns able to discriminate among positive and negative examples and to find new positive examples of the same relation in the corpus. Such new examples are presented to the user for validation and the user feedback is used to refine the patterns or to derive new ones. The process terminates when the user feels that the system has learned to correctly spot the relations. The final patterns are then applied on the whole corpus and the ontology is presented to the user for cleanup.

Cleanup

The cleanup process or post-processing involves a number of steps in order to help the user make the ontology developed by the system coherent. First, the user can visualise the results and edit the ontology directly. They may want to collapse nodes, establish that two nodes are not separate concepts but synonyms, split nodes or move the hierarchical positioning of nodes with respect to each other. Beyond editing the ontology (which in principle could be done in any widely available ontology editor), the user may wish to:

- Add further relations to a specific node
- Ask the learner to find all relations between two given nodes

Label relations discovered in the between given nodes.

The above methodology largely automate the building process. It focuses the expensive user activity on sketching the initial ontology, validating textual examples and the final ontology, while the system performs the tedious and expensive activity of searching a large corpus for knowledge discovery. Moreover the output of the process is not only an ontology, but also a system trained to rebuild and eventually retune the ontology, as the learner is adapted by the user feedback. This simplifies ontology maintenance, a major problem in ontology-based methodologies.

4. Adaptiva[f2]

Adaptiva is a system implementing the methodology above that has been developed as part of the Akt project (Advanced Knowledge Technologies, http://www.aktors.org), an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council. AKT involves the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University (www.aktors.org). Its objectives are to develop technologies to cope with the six main challenges of knowledge management: acquisition, modelling, retrieval/extraction, reuse, publication and maintenance.

The ontology learning process starts with the definition of the draft ontology. **Adaptiva** does not provide any facility for ontology drafting, any of the standard ontology editors (such as protégé) is fine. The ontology is then imported into the system's internal format by using a converter. Concerning corpus definition, **Adaptiva** is based on Gate (www.gate.ac.uk) and the large number of Gate's facilities for corpus definition is available. Lexicalisation of concepts and relations in the ontology are used to retrieve the first set of examples in the corpus. Such examples are presented to the user for validation by using a simple interface shown in Figure 1. This permits the user to specify whether the sentence presented is a positive example or a negative or irrelevant example.

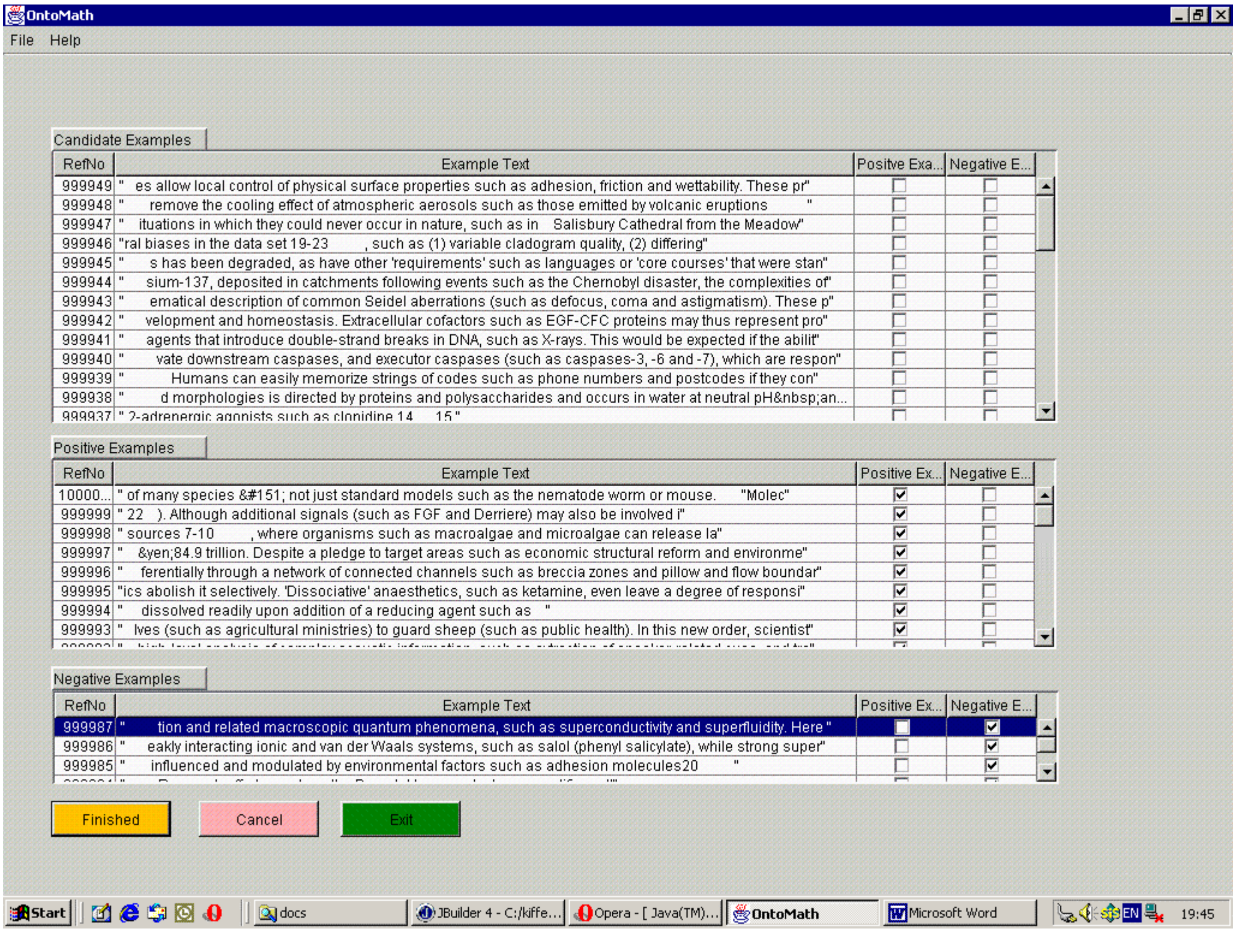
Name of Relation	***Exemplar sentence***	***Positive Example***	***Negative example***
ISA	…countries such as England, France and Italy….	☑	□

Figure 1

The significance of this from the user's perspective is that it is easy and relatively straightforward to validate examples when presented to one in this form. It is much more time consuming to identify such sentences by hand (for example using a KWIC system) let alone performing the lexicographic/editorial task of constructing an ontology on the basis of the exemplars.

The actual complete interface consists of three panes which present i) the examples still to be classified, ii) the examples classified as positive, and iii) those classified as negative (cf. Figure 2). As each example is validated, the user checks one of the two check boxes or leaves the example alone (e.g. because it is too difficult or thought as irrelevant). According to which box is checked the example moves to the positive or negative pane, thereby allowing the user to revise their decision should they wish to do so, and also to look at past choices if needs be[1].

User validation is used to provide the basis for pattern learning. As learner, we use Amilcare (see below), a pattern learning system for information extraction from text also defined within Akt at Sheffield and currently adopted by a number of text annotation tools for the semantic web [Ciravegna 02]. Amilcare uses positive and negative examples to induce patterns. Patterns are generalised so to cover the largest possible number of positive examples in the training corpus avoiding to cover negative ones. Induced patterns are likely to be reliable also on unseen cases.



Candidate Examples

RefNo	Example Text	Positve Exa...	Negative E...
999949	" es allow local control of physical surface properties such as adhesion, friction and wettability. These pr"	□	□
999948	" remove the cooling effect of atmospheric aerosols such as those emitted by volcanic eruptions "	□	□
999947	" ituations in which they could never occur in nature, such as in Salisbury Cathedral from the Meadow"	□	□
999946	"ral biases in the data set 19-23 , such as (1) variable cladogram quality, (2) differing"	□	□
999945	" s has been degraded, as have other 'requirements' such as languages or 'core courses' that were stan"	□	□
999944	" sium-137, deposited in catchments following events such as the Chernobyl disaster, the complexities of"	□	□
999943	" ematical description of common Seidel aberrations (such as defocus, coma and astigmatism). These p"	□	□
999942	" velopment and homeostasis. Extracellular cofactors such as EGF-CFC proteins may thus represent pro"	□	□
999941	" agents that introduce double-strand breaks in DNA, such as X-rays. This would be expected if the abilit"	□	□
999940	" vate downstream caspases, and executor caspases (such as caspases-3, -6 and -7), which are respon"	□	□
999939	" Humans can easily memorize strings of codes such as phone numbers and postcodes if they con"	□	□
999938	" d morphologies is directed by proteins and polysaccharides and occurs in water at neutral pH&nbsp;an...	□	□
999937	" 2-adrenergic agonists such as clonidine 14 15 "	□	□

Positive Examples

RefNo	Example Text	Positive Ex...	Negative E...
10000...	" of many species &#151; not just standard models such as the nematode worm or mouse. "Molec"	☑	□
999999	" 22). Although additional signals (such as FGF and Derriere) may also be involved i"	☑	□
999998	" sources 7-10 , where organisms such as macroalgae and microalgae can release la"	☑	□
999997	" &yen;84.9 trillion. Despite a pledge to target areas such as economic structural reform and environme"	☑	□
999996	" ferentially through a network of connected channels such as breccia zones and pillow and flow boundar"	☑	□
999995	"ics abolish it selectively. 'Dissociative' anaesthetics, such as ketamine, even leave a degree of responsi"	☑	□
999994	" dissolved readily upon addition of a reducing agent such as "	☑	□
999993	" lves (such as agricultural ministries) to guard sheep (such as public health). In this new order, scientist"	☑	□

Negative Examples

RefNo	Example Text	Positive Ex...	Negative E...
999987	" tion and related macroscopic quantum phenomena, such as superconductivity and superfluidity. Here "	□	☑
999986	" eakly interacting ionic and van der Waals systems, such as salol (phenyl salicylate), while strong super"	□	☑
999985	" influenced and modulated by environmental factors such as adhesion molecules20 "	□	☑

Figure 2

When the learning process is finished, Amilcare applies the induced patterns on the unseen corpus and returns a number of new examples to be classified by the user. The user is never requested to classify an already classified example. This iterative process may continue until the user is satisfied, which in this context means that a sufficiently

[1] The validation process results in a three-way classification of each sentence: valid, invalid or irrelevant. The Machine Learning procedure learns from the positive and the negative exemplars but NOT from the irrelevant one.

high proportion of exemplars are correctly classified by the system automatically (i.e. the proposed examples in the first window tend to be largely correct).

4.1. Using a learning algorithm

The methodology described above is generic in that it is not tied to one specific Machine Learning algorithm or approach. The precise methodology by which rules are learned from the examples tagged by the user is irrelevant from the user's perspective. In **Adaptiva**, we have integrated Amilcare, a tool for adaptive Information Extraction from text (IE) designed for supporting active annotation of documents for the Semantic Web. Amilcare is the perfect support for the task above because it is able to induce patterns without requiring any knowledge of NLP and uses an ontology as a basis for learning.

Amilcare performs IE by enriching texts with XML annotations, i.e. the system marks the extracted information with XML annotations. The only knowledge required for porting Amilcare to new applications or domains is the ability of manually annotating the information to be extracted in a training corpus. No knowledge of Human Language technology is necessary. Adaptation starts with the definition of a tagset for annotation possibly organised as an ontology where tags are associated to concepts and relations. Then users have to manually annotate a corpus for training the learner. An annotation interface is to be connected to Amilcare for annotating texts using XML mark ups. As mentioned Amilcare has been integrated with a number of annotation tools so far, including MnM (Domingue *et al.* 02), Ontomat (Handschuh et al. 02), Melita (Ciravegna *et al.* 02) and the Gate annotation tool (www.gate.ac.uk). For example the annotation interface in Ontomat is used to annotate texts in a user-friendly manner. Ontomat automatically converts the user annotations into XML tags to train the learner. Amilcare's learner induces rules that are able to reproduce the text annotation.

Amilcare can work in two modes: training, used to adapt to a new application, and extraction, used to actually annotate texts. In both modes, Amilcare first of all preprocesses texts using Annie, the shallow IE system included in the Gate package ([9], www.gate.ac.uk). Annie performs text tokenization (segmenting texts into words), sentence splitting (identifying sentences) part of speech tagging (lexical disambiguation), gazetteer lookup (dictionary lookup) and named entity recognition (recognition of people and organization names, dates, etc.).

When operating in training mode, Amilcare induces rules for information extraction. The learner is based on $(LP)^2$, a covering algorithm for supervised learning of IE rules based on Lazy-NLP [10] [11]. This is a wrapper induction methodology [12] that, unlike other wrapper induction approaches, uses linguistic information in the rule generalization process. The learner starts inducing wrapper-like rules that make no use of linguistic information, where rules are sets of conjunctive conditions on adjacent words. Then the linguistic information provided by Annie is used in order to generalise rules: conditions on words are substituted with conditions on the linguistic information (e.g. condition matching either the lexical category, or the class provided by the gazetteer, etc. [11]). All the generalizations are tested in parallel by using a variant of the AQ algorithm [13] and the best {\it k} generalizations are kept for IE. The idea is that the linguistic-based generalization is used only when the use of NLP information is reliable or effective. 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. Lazy NLP-based learners learn which is the best strategy for each information/context separately. For example they may decide that using the result of a part of speech tagger is the best strategy for recognizing the speaker in seminar announcements, but not to spot the seminar location. This strategy is quite effective for analyzing documents with mixed genres, quite a common situation in web documents [14].
The learner induces two types of rules: tagging rules and correction rules. A tagging rule is composed of a left hand side, containing a pattern of conditions on a connected sequence of words, and a right hand side that is an action inserting an XML tag in the texts. Each rule inserts a single XML tag, e.g. `</speaker>`. This makes the approach different from many adaptive IE algorithms, whose rules recognize whole pieces of information (i.e. they insert both `<speaker>` and `</speaker>`[7]), or even multi slots [15]. Correction rules shift misplaced annotations (inserted by tagging rules) to the correct position. They are learnt from the mistakes made in attempting to reaanotate the training corpus using the induced tagging rules. Correction rules are identical to tagging rules, but (1) their patterns match also the tags inserted by the tagging rules and (2) their actions shift misplaced tags rather than adding new ones. The output of the training phase is a collection of rules for IE that is associated to the specific scenario.
When working in extraction mode, Amilcare receives as input a (collection of) text(s) with the associated scenario (including the rules induced during the training phase). It preprocesses the text(s) by using Annie and then it applies its rules and returns the original text with the added annotations. The Gate annotation schema is used for annotation [9].

In **Adaptiva** Amilcare is used in the following way: positive and negative examples as provided by the user are transformed into a training corpus where XML annotations are used to identify the occurrence of relations in positive examples. The rest of the corpus (i.e. everything it is not annotated) is considered a negative example of the specific relation. The learner is then launched and patterns are induced and generalised using $(LP)^2$. Patterns are tested on the training corpus and the best most generic patterns are retained. Such patterns are then applied to the unseen corpus to retrieve other examples. From Amilcare's point of view the task of ontology learning is transformed into a task of text annotation: the examples are transformed into annotations and annotations are used to learn how to reproduce such annotations. When unseen annotated examples are returned to the user, the annotation is removed and the example is presented by the user. We are currently considering keeping the actual annotation in order to help readability of examples.

5. Conclusion and Future Work

We have presented a novel model of user-system interaction for the purposes of ontology building specifically in the context of knowledge management. This work implements to a larger degree the ideas first proposed by Hearst and built on by Morin. We believe that this is just a first step in a new direction of using NLP (and in particular IE) for user-centred ontology building that could potentially considerably impact the way in which ontologies are built for real world applications.

Future work has two directions. First, the integration of other sources of data derived from text using a number of techniques to build seed ontologies and establish the existence of association between terms. This will permit an ontology editor to identify new concepts and new associations between existing concepts.

Secondly, we are currently evaluating in detail the qualitative and ergonomic aspects of the system so as to establish exactly what the benefits are, to what degree and how the system can be further improved for the user. It is difficult to benchmark complex systems such the one presented above but there are number of criteria to help determine how the system can be improved (Brewster 2002).

Acknowledgements

The current work has been carried on in the framework of the AKT project (Advanced Knowledge Technologies, http://www.aktors.org), an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01). AKT involves the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University (www.aktors.org). AKT is a multimillion pound six year research project that started in 2000. Its objectives are to develop technologies to cope with the six main challenges of knowledge management: acquisition, modelling, retrieval/extraction, reuse, publication and maintenance.

Amilcare is based on Gate (www.gate.ac.uk). Thanks to the Gate group of the University of Sheffield for providing the Annie system and for help in integrating Amilcare and Annie.

References

Alashawi, H. (1987) *Memory and Context for Language Interpretation* Cambridge, England: Cambridge University Press

Berners-Lee, T., J. Hendler, O. Lassila, (2001) *The Semantic Web*, in Scientific American Issue 501 (http://www.sciam.com/2001/0501issue/0501berners-lee.html)

Brewster, C. (2002) Techniques for Automated Taxonomy Building: Towards Ontologies for Knowledge Management, in *Proceeding of the 5th Annual CLUK Research Colloquium*, Leeds, pp.

Brewster, C., F. Ciravegna, and Y. Wilks, (2001) Knowledge Acquisition for Knowledge Management: Position Paper, in *Proceeding of the IJCAI-2001 Workshop on Ontology Learning* held in conjuction with the 17th International Conference on Artificial Intelligence (IJCAI-01), Seattle, August, 2001

Brown, Peter F.,Vincent J. Della Pietra, Petere V. DeSouza, Jenefer C. Lai, Robert L. Mercer, (1992) *Class-based n-gram models of natural language*, Computational Linguistics, 18, 467-479

Ciravegna, F. (2001a) “Challenges in Information Extraction from Text for Knowledge Management”, *IEEE Intelligent Systems and Their Applications*, November 2001.

Ciravegna, F.(2001b) "(LP)^2, an Adaptive Algorithm for Information Extraction from Web-related Texts" in Proceedings of the *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining* held in conjunction with the 17th International Conference on Artificial Intelligence (IJCAI-01), Seattle, August, 2001

Ciravegna, Fabio, (2001c), *Adaptive Information Extraction from Text by Rule Induction and Generalisation* in “Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)” , Seattle, August 2001.

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

Fensel, D., F. van Harmelen, I. Horrocks, D.L. McGuiness, P.F. Patel-Schneider, (2001) *OIL: An Otology Infrastructure for the Semantic Web*, IEEE Intelligent Systems, 16: pp.38-45

Finkelstein-Landau, Michal, Emmanuel Morin, (1999) *Extracting Semantic Relationships between Terms: Supervised vs. Unsupervised Methods*, in "Proc. International Workshop on Ontological Engineering on the Global Information Infrastructure", 71-80, Dagstuhl Castle, Germany,

Gomez-Perez, A., (1999) *Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases*, in "Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop", Banff, Alberta, Canada, 16-21 October 1999

Grefenstette, Gregory, (1994), *Explorations in Automatic Thesaurus Discovery*, Kluwer.

Guarino, N. and Welty, C. 2000. *Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis*. In H. Werner (ed.) ECAI-2000: The European Conference on Artificial Intelligence. IOS Press, Berlin, Germany: 219-223.

Guthrie, L., B. Slator, Y. Wilks, and R. Bruce (1990) Is there content in empty heads? *Proceeding of the 13th International Conference on Computational Linguistics (COLING-90)*, Helsinki, vol 3, pp.138-143.

Hearst, M.A., (1992) *Automatic Acquisition of Hyponyms from Large Text Corpora*, in "Proc. of COLING 92", Nantes

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

Maynard, D., V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva and Y. Wilks: "Architectural Elements of Language Engineering Robustness", Journal of Natural Language Engineering -- Special Issue on Robust Methods in Analysis of Natural Language Data, 2002, forthcoming.

Mickalski, R. S., I. Mozetic, J. Hong and H. Lavrack: *The multi purpose incremental learning system AQ15 and its testing application to three medical domains*', in Proceedings of the 5th National Conference on Artificial Intelligence, Philadelphia: Morgan Kaufmann publisher.

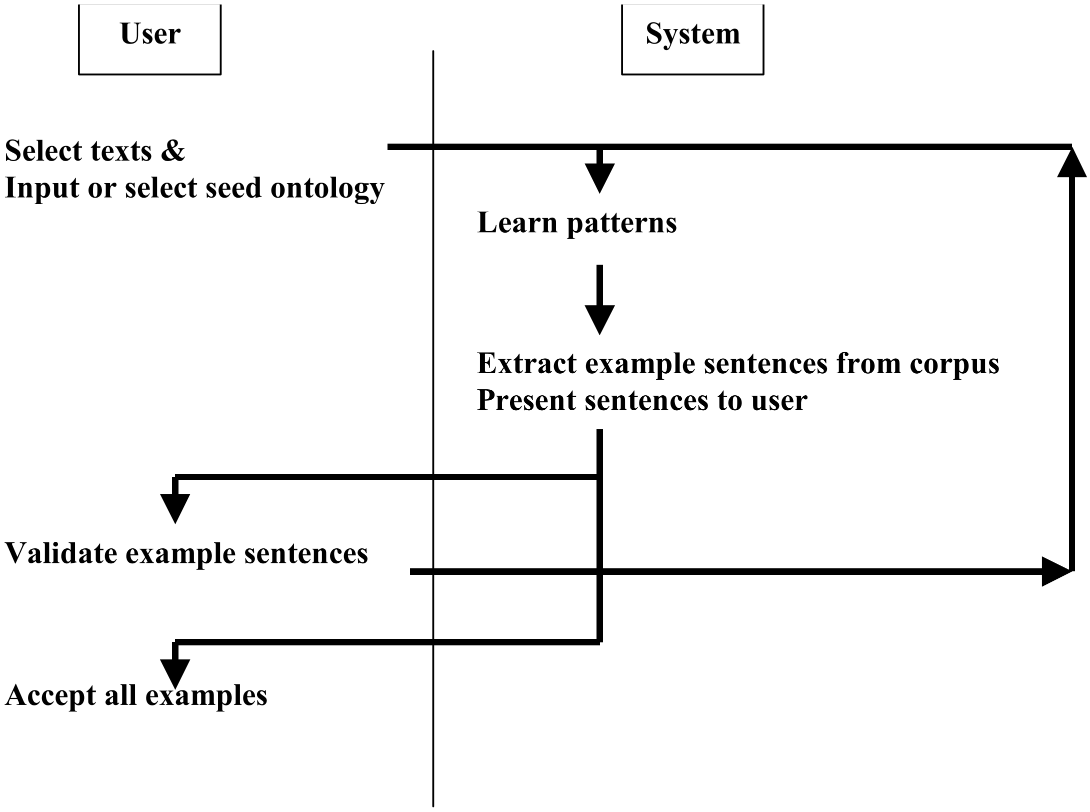
Morin, Emmanuel, (1999a), *Des Patrons lexico-syntaxiques pour aider au depouillement terminologique*, Traitment Automantique des Langues, 40:1, 143-166,

Morin, Emmanuel, (1999b), *Using Lexico-Syntactic patterns to Extract Semantic Relations between Terms from Technical Corpus*, in "Proc. of TKE 99", 268-278, Innsbruck, Austria,

Sanderson, Mark, and Bruce Croft, (1999) *Deriving concept hierarchies from text*, in "Proceedings of the 22nd ACM SIGIR Conference", pp. 206-213,

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

Figure


User
System
Select texts &
Input or select seed ontology
Learn patterns
Extract example sentences from corpus
Present sentences to user
Validate example sentences
Accept all examples