User involvement in customizing adaptive Information Extraction: position paper

Fabio Ciravegna[#] and Daniela Petrelli[*]

Introduction

In the last years, research on adaptive Information Extraction from text (IE) has largely focused on algorithms and systems adaptable to new Web-related applications/scenarios by users with analyst's knowledge, i.e. knowledge on the domain/scenario only, [Kushmerick 1997], [Califf 1998], [Muslea 1998], [Freitag 1999], [Soderland 1999], [Freitag 2000], [Ciravegna 2001]. Successful commercial products have been created and there is an increasing interest on IE in the Internet market. The more the focus is on the user, the more the need for user-specific tools arises. Most of the current approaches are based on an adaptation phase in which the user provides a set of texts with the relevant information highlighted or with associated filled templates. Tagging is just one part of the adaptation task, though, in building real world applications. Adaptation as a one-way process (from tagged examples to rules) is unlikely to provide optimized results for specific users, as different uses will require different types of results (e.g., high recall in some cases, high precision in others). There is the necessity, we believe, to fully support users during the whole adaptation process so to maximize effectiveness and appropriateness of the final application, and to minimize the burden of system adaptation. In this paper we discuss requirements for user involvement in application development in Amilcare, a system for adaptive IE.

1. Amilcare

Amilcare is a system for adaptive IE from Internet related texts that is under development as part of the AKT project, an important project for research on the future of knowledge management. Amilcare aims to provide fully customisable IE for knowledge management purposes. It is based on the **(LP)^2** algorithm for adaptive IE [Ciravegna2001]. Its interface will support users in the whole IE application development process, maximizing effectiveness and appropriateness of the final application and minimizing the burden of system adaptation from the user side. The intended user is a person expert in the domain, with limited analyst's skills and with no knowledge on IE or NLP. Most of the requirements for Amilcare have been derived by our experience in designing and delivering $_{Learning}$Pinocchio, a commercial system for adaptive IE [Ciravegna 2001].

The proposed application development cycle in Amilcare includes the following steps: scenario design, system adaptation, results validation. These three steps are iterated until a satisfying application is reached and delivered; they are discussed in the rest of this section.

1.1 User wishes and Scenario design

The application design starts with an initial analysis of user's needs or wishes. Moving from user wishes to the actual IE scenario design is non-trivial. There may be a gap between what information the user needs, what information the texts contain and what the system can actually extract. Therefore it is very important, especially when working with naïve users, that the system interface helps recognizing such discrepancies, first of all by clearly conveying what type of information the system is able to extract. An effective interface should help identifying the type of information the texts contain by forcing the user into the right paradigm of scenario design. In Amilcare we plan to guide users step by step in defining the scenario in a way similar to what [Yangarber00] proposes for pattern learning, i.e.:

1. The user highlights a group of relevant sentences from the untagged corpus;
2. The system retrieves other relevant sentences from an untagged corpus by using some regularities found in the provided examples. The user validates the system defined examples;
3. The new sentences are used to identify new regularities and for retrieving some more sentences. This process is iterated until no now relevant examples are found.
4. The information present in the previously selected set of sentences is highlighted by the user in different colors representing different fillers.

This design mechanism is mainly suitable for inexperienced users since they are driven by the system toward an exhaustive collection of representative cases. This process is likely

[#] Department of Computer Science, the University of Sheffield,
Regent Court, 211 Portobello Street, S1 4DP Sheffield, UK,
Email: F.Ciravegna@dcs.shef.ac.uk
[*] Istituto per la Ricerca Scientifica e Tecnologica, via Sommarive
18, 38050, Povo, Trento, Italy. Email: petrelli@irst.itc.it

to reduce the risk to base the application development on a set of sentences that do not completely cover the final domain. For this reason even skilled users can find this approach worthwhile.

Once the scenario is properly designed, there is the need of adapting the system. This point is discussed in the following section.

1.2 Training the system

Ideal users for an adaptive IE system are expert analysts: they are good at defining scenarios and identifying relevant information (e.g. by filling templates) and have generally clear ideas about the IE task to be performed. For these people, the interface should support complex and detailed corpus tagging (e.g. template-based tagging, inclusive of co-references among events). The largest part of the potential users are naïve and they need specific support, as they may find difficult to manipulate IE-related concepts, e.g. the standard template concepts. Highlighting information in different colors is generally considered simpler than template filling and mainly avoids the problem of identification of co-references (necessary for template-based IE) that is often very difficult for a layman. In general tagging-based interfaces, such as Mitre's Alembic, have proven to be quite effective for both complex analyst's tagging and naïve user tagging and have become a standard among many participants in the MUC conferences and in real world adaptive IE systems.

1.2.1 Active Learning

We believe that efficient and effective corpus tagging can be obtained by using active learning [Soderland99], [Califf98]. In active learning an initial tagging is provided for a limited number of examples. The number of tagged examples is then increased by the system by identifying in an untagged corpus both a number of similar texts (on which the system induced rules are effective) and other sets of texts for which the system is not able to extract reliable information. This approach supports accommodation of task sharing between user and system at the best. First of all because validating extracted information is much simpler task than tagging untagged texts (and therefore less error prone and more efficiently performed). Secondly this technique reduces the number of texts to be manually tagged, because it focuses the slow and high cost user activity on examples the system is not able to tag by itself, while avoiding requiring tagging texts on which it has already reached a satisfying effectiveness. Finally active learning helps concentrating on information deviating from the standard the user has in mind, therefore providing useful hints for scenario revision, if necessary (in our experience scenario revision is often necessary to cope with unexpected phenomena that emerge from tagging).

We plan to provide Amilcare with some active learning capabilities. In order to perform active learning the system must be able to evaluate how good its performances are on an untagged text. Extensions to active learning techniques such as co-testing ([Muslea 00]) can be used. In co-testing different groups of rules are derived that cover the same cases. When those groups stop recognizing same portions of input in new texts, a deviation in the information format can be identified and texts can be presented to users for further tagging and training.

1.2.2 Training corpus selection

Training corpus selection for real world applications can be a delicate issue. Inexperienced users often tend to provide training corpora that are not representative enough of the texts the system will find when operating in the final application. The corpus may be unbalanced with respect to genres (e.g. emails could be underrepresented in a corpus about communication monitoring), or present non-updated features (e.g. a corpus of old web pages designed for old versions of browsers may be provided) or even show peculiar regularities due to wrong selection criteria. For example in designing an application on financial news the user selected a corpus made of news issued on three consecutive days only: he claimed that three days should have covered a sufficient number of issues. Unfortunately the corpus resulted not to be representative enough, if not definitely unbalanced wrt the covered topics. For example many news referred to the quotation of one specific company in the stock exchange and comments and reaction related to that quotation. Moreover there was a number of news related to some specific fiscal deadline to come in those days. The result of training was a largely ineffective set of rules that left the user dissatisfied with the final application. We believe therefore that it is necessary to provide some tools for validating the training corpus with respect to an (hopefully big) untagged corpus that may be available. The untagged corpus can be mined in order to verify how much the training corpus is representative. One possibility concerns the formal comparison of training and untagged corpus. [Kilgarriff 2001] proposes heuristics for discovering differences in text genres among corpora. Average text length, distributions of HTML tags and hyperlinks in web pages, average frequency of lexical classes in texts (e.g. nouns), etc. can be relevant indicators of corpus representativeness and can be used to warn inexperienced users that some training corpora can be not representative enough of the whole corpus. Even the detection of an excess of regularity in the training corpus can be a good indicator of an unbalanced corpus selection, e.g. if a high percentage of fillers for some slots is the same string. Another strand of corpus validation that we plan to use, will be mentioned in the result validation phase below.

1.3 Result validation

Users should be enabled to evaluate results from both a quantitative and qualitative point of view. A test corpus with associated expected results will be the basis for evaluation in Amilcare. The system can be run and statistics on both effectiveness and accuracy can be presented to the user, together with details on correct matches and mistakes. The MUC scorer [Douthat 1998] provides such information.

The user must be enabled to let the system know how much she likes such results, in order to eventually modify the system behavior according to her needs (e.g. more precision and less recall). In case of occasional or inexperienced users, the issue arises of avoiding the use of technical or numerical concepts (such as precision and recall). This requires the ability from the interface of bridging the user' s qualitative vision (" you are not capturing enough information") with the numerical concepts the system is able to manipulate (e.g. moving error thresholds in order to obtain higher recall).
Validation of results with respect to the final application is another important aspect. The induced grammars can be used to analyze an untagged corpus in order to identify groups of texts on which the system does not perform properly. Such areas are very likely to be representative of a type of text insufficiently represented in the training corpus, or, more generally, to be an indication that the training corpus was not representative enough as mentioned above.

1.4 Delivering the Application

The last step in the development cycle is application delivery. The system should be able to provide a runtime version that does not include the development environment (e.g. the interface), but only the core of the IE engine. Such runtime should include some corpus monitoring facilities, though. One of the risks in highly changing environments such as the Internet is that information (e.g. web pages) can change format in a very short time, and the system must be able to detect such changes [Muslea 2000]. The same techniques mentioned above for testing corpus representativeness can be use to identify changes in the information structure or test type.

2 Conclusion

In this paper we have analysed a number of requirements for effective human-computer interaction for adaptive IE-based applications. We are currently defining Amilcare's interface by addressing the issues mentioned above. Amilcare is developed as part of the AKT project, a multi million pounds project funded by EPSRC in the UK for research on the future of knowledge management. Partners in the projects are the University of Sheffield, Southampton, Edinburgh, Aberdeen and Open University.

References

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

[Ciravegna 2001] Fabio Ciravegna, 'Adaptive Information Extraction from Text by Rule Induction and Generalisation' *in of the 17th International Joint Conference On Artificial Intelligence (IJCAI2001)*, Seattle, August, 2001.

[Douthat 1998] Aaron Douthat, `The message understanding conference scoring software user's manual', in *the 7th Message Understanding Conference*, (www.muc.saic.com)

[Freitag 1999] Dayne Freitag and Andrew McCallum: `Information Extraction with HMMs and Shrinkage', *AAAI-99 Workshop on Machine Learning for Information Extraction*, Orlando, 1999, (www.isi.edu/~muslea/RISE/ML4IE/)

[Freitag 2000] Dayne Freitag and Nicholas Kushmerick, `Boosted wrapper induction', in F. Ciravegna, R. Basili, R. Gaizauskas (eds.) *ECAI2000 Workshop on Machine Learning for Information Extraction*, Berlin, 2000, (www.dcs.shef.ac.uk/~fabio/ecai-workshop.html)

[Kilgarriff 2001] A. Kilgarriff, 'Comparing Corpora', to appear in Corpus Linguistics.

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

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

[Muslea 2000] I. Muslea, S. Minton, and C. Knoblock, 'Co-testing: selective sampling with redundant views' in *Proc. of the 17th National Conference on Artificial Intelligence, AAAI-2000*.

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

[Yangarber 2000] Roman Yangarber, Ralph Grishman, Pasi Tapanainen and Silja Huttunen: ``Automatic Acquisition of Domain Knowledge for Information Extraction" In *Proc. of COLING 2000: The 18th International Conference on Computational Linguistics*, Saarbrücken, 2000.