

Document Annotation via Adaptive Information Extraction

Categories and Subject Descriptors: Information Extraction

Keywords: Information Extraction from Text, Machine Learning, Knowledge Management

1. INTRODUCTION

The traditional process of document annotation for knowledge identification and extraction in Knowledge Management (KM) is complex and time consuming, as it requires manual annotation by domain experts. In the typical scenario a domain expert:

1. builds an ontology describing the application domain;
2. annotates a number of texts in order to identify instances of elements (i.e., concepts and relations) in the texts.

There is a strong interest in Text Mining technologies (and in particular in Human Language-based Technologies), for reducing the burden of text annotation for KM [1]. In this paper we show how Information Extraction from texts (IE) can provide support for document enrichment and make the text annotation process more effective and efficient.

The main challenge to be addressed by IE researchers in this framework is portability of IE systems to new applications with no knowledge of HLT. As a matter of fact, in terms of IE each annotation task (e.g. tagging texts about failures in cars in order to identify faults and involved car parts) requires to port the IE system to a new application domain. IE is just one of the many technologies required by complex KM environments: wide use of IE tools will come only when the definition of such application domain will not require any specific IE skill apart from notions of KM. Moreover there is the need to port across different text types without major recoding of system resources, as documents in KM can range from free texts (technical reports, newspaper-like texts) to (semi)structured marked up texts (e.g. partially marked-up or even highly structured XML/HTML documents) and even a mixture of them [3].

2. AMILCARE

Amilcare is a tool for adaptive IE for supporting active annotation of documents for KM. Its design addresses the portability requirement above: to be adapted to new applications the system just requires the ability of annotating documents. Once adapted, the system can be used either for unsupervised annotation or as a support for human annotation. Amilcare is based on (LP)², a supervised learning algorithm for IE [4, 5]. Adaptation starts with the definition of a tagset for annotation possibly organized as an ontology where tags are associated to concepts and relations. This is consistent with the current practice in KM, where ontologies are used for describing application domains. Then users have to manually annotate a corpus for training the learner. Amilcare provides an easy-to-use interface for corpus annotation where different colors are associated to different tags. Annotations are inserted by first selecting a tag from the ontology and then identifying the text area to annotate with the mouse. Differently from similar annotation tools [6, 7], Amilcare actively helps in annotating the training corpus. While users annotate texts, Amilcare learns how to reproduce the inserted annotation by running in the background its learner. Moreover

the induced rules are silently applied to each new text and their results are compared with the user annotation. When its rules reach a (user-defined) level of accuracy, Amilcare presents each new text with a proposed preliminary annotation derived by the rule application. In this case users have just to revise the proposed annotation by correcting mistakes and adding missing annotations. User corrections are inputted back to the learner for retraining. This technique (based on active learning [8]) focuses the slow and expensive user activity on uncovered cases, avoiding requiring annotating cases where a satisfying effectiveness is already reached. Moreover validating extracted information is a much simpler task than tagging bare texts (and also less error prone), speeding up the process considerably. At the end of the corpus annotation process, the system is trained and the application can be delivered.

2.1 USERS, APPLICATIONS AND SKILLS

The only knowledge required for building new applications is the ability to annotate documents, i.e. no knowledge of IE is necessary. Actually, Amilcare is designed to accommodate the needs of different user types. While naïve users can build new applications without delving into the complexity of HLT, IE experts are provided with a number of facilities for tuning the final application. Induced rules can be inspected, monitored and edited to obtain some additional accuracy, if needed. The interface also allows balancing precision (P) and recall (R). The system is run on an annotated unseen corpus and users are presented with statistics on accuracy, together with details on correct matches and mistakes. Retuning the P&R balance does not generally require major retraining. Facilities for inspecting the effect of different P&R balances are provided. Although the current interface for balancing P&R is designed for IE experts, we have plans for enabling also naïve users [10].

2.2 ADAPTING TO TEXT TYPES

While most of the IE literature has focused on IE from free newspaper-like texts, KM applications, with their composite web-based scenarios, require portability across text types (see above). Linguistically-based methodologies used for free texts (e.g. [12]) can be difficult to apply or even ineffective on highly structured marked up documents. They are not able to cope with the extralinguistic structures (e.g. tags, document formatting) used to convey information in such documents. On the other hand, wrapper-like algorithms designed for highly structured documents are largely ineffective on unstructured texts given their inability to overcome data sparseness due to linguistic variation [5]. Amilcare is based on Lazy-NLP, a methodology that attempts to fill the gap between linguistic and wrapper-like approaches [4]. A Lazy-NLP learner learns the best (most reliable) level of linguistic analysis useful (or effective) for a specific IE task by mixing linguistic and shallow strategies. The learner starts inducing wrapper-like rules that make no use of linguistic information. Then it progressively adds linguistic information to its rules, stopping when the use of NLP information becomes unreliable or ineffective. 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.

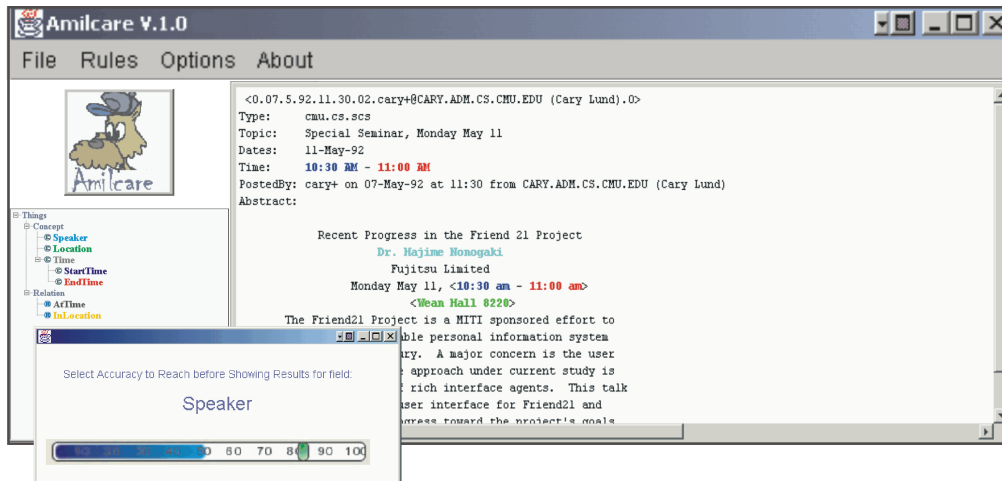


Figure 1: The tagging Interface: left is the tagset, right the text. Note the dialog box for tailoring active learning (the blue bar represent the reached accuracy, the slide bar the wanted accuracy).

Lazy NLP-based learners learn which is the best strategy for each information/context separately. For example they may decide that using parsing is the best strategy for recognising the speaker in seminar announcements, but not to spot the seminar location. This is quite effective for analysing documents with mixed genres, a common situation in web documents [3].

2.3 ADAPTING GENERIC RESOURCES

Even if Lazy-NLP learners use generic NLP modules and resources not to be modified for specific application needs, Amilcare provides a way of retuning some types of generic IE resources. For example Named Entity Recognisers (NERs) can become quite ineffective if used to analyse reports of trials from the 19th century because location names had different styles at the time, or because a number of first names for people have now disappeared. In this case it is either possible to let the learning algorithm decide how and when the NER is reliable (in a pure Lazy-NLP perspective), or it is possible to use Amilcare to learn how to patch the NER's results. Even naïve users can accomplish this by using the same interface provided for annotating corpora. In this case the generic resource is used to provide the initial preliminary tagging of texts to be corrected by users. User corrections are inputted back to Amilcare that learns how to reproduce such correction, i.e. how to cover missing cases and to shift misplaced tags. We are currently modifying the (LP)² algorithm so to remove spurious annotations.

3. CONCLUSION

Amilcare is a tool for adaptive IE designed for KM purposes that requires no IE skills for porting to new applications. It is able to cope with different texts types without requiring major recoding of resources. It easily integrates with the usual manual document annotation process. Initially it appears as one of the usual KM annotation tools. When its rules become reliable it automatically starts helping in the annotation process. At some point the user can decide either to leave Amilcare to continue the annotation process (unsupervised annotation) or to continue to manually annotate the documents letting Amilcare suggesting a draft annotation (supervised annotation). Either way the result is a more efficient and effective process. Amilcare uses cutting edge IE technology: the (LP)² algorithm obtains excellent results in scientific experiments [4] and was also used to build a number of real

world applications [4]. Concerning suitability for KM purposes: on the one hand Amilcare has been successfully integrated in the AKT2 architecture and is used to build experimental applications within the AKT project (www.aktors.org). On the other hand Amilcare's learning kernel has been integrated in two tools for Knowledge Management: Mnm (the Open University) and Ontomat, (University of Karlsruhe).

4. REFERENCES

- [1] M. Maybury (ed.): Proc. of the 2001 EACL/ACL Workshop on Human Language Technology and Knowledge Management, at the 39th meeting of the ACL, July 6-11, 2001, Toulouse, France
- [2] F. Ciravegna: "Challenges in Information Extraction from Text for Knowledge Management", *IEEE Intelligent Systems and Their Applications*, November 2001.
- [3] F. Ciravegna: "Adaptive Information Extraction from Text by Rule Induction and Generalisation" in *Proc. of 17th International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, August 2001."
- [4] F. Ciravegna: "(LP)², an Adaptive Algorithm for Information Extraction from Web-related Texts" in *Proc. of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, August, 2001
- [5] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson and M. Vilain *Mixed-initiative development of language processing systems*. In Proc. of the Fifth Conference on Applied Natural Language Processing, Washington, 1997.
- [6] H. Cunningham, D. Maynard, V. Tablan, C. Ursu, K. Bontcheva: "Developing Language Processing Components with GATE", www.gate.ac.uk
- [7] S. Soderland, D. Fisher, J. Aseltine, Wendy Lehnert: "CRYSTAL: Inducing a Conceptual Dictionary", in Proc. of the 14th International Joint Conference on Artificial Intelligence IJCAI '95
- [8] F. Ciravegna and D. Petrelli: "User Involvement in Adaptive Information Extraction: Position Paper" in *Proc. of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, August, 2001
- [9] 7th Message Understanding Conference Proc., www.itl.nist.gov/iaui/894.02/related_projects/muc/