Mining Web Sites Using Adaptive Information Extraction

Alexiei Dingli and Fabio Ciravegna and David Guthrie and Yorick Wilks
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street,
S1 4DP Sheffield, UK

1 Introduction

Adaptive Information Extraction systems (IES) are currently used by some Semantic Web (SW) annotation tools as support to annotation (Handschuh et al., 2002; Vargas-Vera et al., 2002). They are generally based on fully supervised methodologies requiring fairly intense domain-specific annotation. Unfortunately, selecting representative examples may be difficult and annotations can be incorrect and require time. In this paper we present a methodology that drastically reduce (or even remove) the amount of manual annotation required when annotating consistent sets of pages. A very limited number of user-defined examples are used to bootstrap learning. Simple, high precision (and possibly high recall) IE patterns are induced using such examples, these patterns will then discover more examples which will in turn discover more patterns, etc.

The key feature that enables such bootstrapping is the *Redundancy* on the Web. Redundancy is given by the presence of multiple citations of the same facts in different superficial formats and is currently used for several tasks such as improving question answering systems (Dumais et al., 2002) and performing information extraction using machine learning (Mitchell, 2001). When known information is presented in different sources, it is possible to use its multiple occurrences to bootstrap recognisers that when generalised will retrieve other pieces of information, producing in turn more (generic) recognisers. In our model redundancy of information is increased by using preexisting services (e.g. search engines, digital libraries). This improves the effectiveness of bootstrapping.

Another typical feature of Web pages that we exploit for learning is *document formatting*: HTML and XML pages often contain formatting directives (tables, lists, etc.) that group identical or related information. Identifying such formatted areas and their content can be very useful. For example, a structure listing some known names can be used to discover other names if it is possible to generalise over the regularity of the list.

In the rest of the paper we will present the details of our methodology as implemented in the Armadillo System, using an application of IE from Computer Science Web sites as a matter of exemplification.

2 Methodology in action

The task is to mine Computer Science Department Web sites in order to extract data about people and to discover communities of practice (who works with whom) and their evolution in time. We define the task as: (1) the identification of lists of people who work in a department, (2) the extraction of personal data from personal web pages (position, email address, telephone number, etc.); (3) the identification of groups of people working together by monitoring (a) publication lists (and publication date of each work to trace the evolution in time); (b) research projects they are involved in. Until now, manually site-specific handcrafted wrappers have been used to perform this task (Shadbolt, 2002). Our aim is to define a largely automatic method that does not require any user intervention either in the form of wrapper writing, or in the form of extensive examples annotation.

In Armadillo, we use an **incremental strategy**: we start from a handful of simple examples provided by a user (e.g. a list of project names and a CS Web site). As mentioned above, such names will be used to discover other names, possibly the full list. Large lists will be used to bootstrap identification of other more sophisticated information (e.g. identifying project pages and involved people) that will in turn be used to bootstrap learning of more sophisticated information (e.g. communities of practice) and so on, until the whole (or the large part of) information is identified and extracted. At each stage, we use **a number of strategies** for recovering the desired information. We apply the simplest and more reliable strategies first, resorting to more sophisticated or less reliable strategies only when necessary. For example we exploit available tools such as pre-existing classifiers, digital libraries, search engines, etc. where available and reliable, otherwise we create domain-specific IE engines, or, if not possible, we use weaker methods such as keyword matching. In the rest of the section we will focus on the different steps in the CS application.

2.1 Identifying lists of relevant names

The simplest way to identify personal pages is to use specifically trained classifiers (e.g. http://www-2.cs.cmu.edu/~webkb/) when available and provided that they have high accuracy. If they are not available (as in the case of project pages classifiers), or accurate enough, or the number of returned pages is not satisfying, more sophisticated strategies must be used. In our case no classifiers are available, so we start identifying names of people using an existing Named Entity Recogniser (NERC). NERCs are generally very reliable ($>$ 90%) in identifying people's names. These names are further checked using publicly available services such as CiteSeer (citeseer.com) in order to further guarantee the validity of a name. Also, CiteSeer will provide additional information for a particular person (e.g. coauthors, publications) which the system stores for further use. For project names there are no NERCs available, so we induce an ad-hoc recognizer using some minimal user input. We use a short user-defined list of project names and a set of pages where such names are likely to appear (i.e. an unannotated CS website and five names of relevant projects). The system automatically annotates the occurrences of such names on the pages and uses an adaptive IE algorithm to induce a small number of high-precision high-recall patterns, which will produce other annotations and derive further patterns. The redundancy on the Web ensures that we will find a reasonable number of examples to train on. This process is difficult to control in principle, but using high-precision high-recall settings for the induction algorithm (e.g. very strict error thresholds) and limiting the number of learning cycles, it is possible to derive a set of patterns that are generic and reliable enough to work on **any** CS sites (so they are created once for all). Such lists represent just provisional information; the presence of limited noise is not a problem.

In order to derive further names, we use the list of derived projects' and people's names to query a search engine in order to look for pages containing a high quantity of such names, possibly organised in highly formatted areas such as structured lists. For example most CS sites contain pages dedicated to staff or project listing. On such structured areas, it is possible to use the known names to annotate examples and induce wrappers (Kushmerick et al., 1997). Wrappers are IE systems that rely heavily on document formatting. These wrappers identify more complete lists of names, eventually used to bootstrap further searching and learning, if necessary (using only highly reliable examples to avoid noise).

2.2 Retrieving personal and projects' pages

Lists of names are used to identify personal or project pages by using hyper-links directly associated to names. To check the validity of those links or if the pages for a number of people/projects are not found, it is possible to use publicly available services such as Google (www.google.com) or HomePageSearch (http://hpsearch.uni-trier.de/), restricting the search to pages in the specific site at hand.

In summary we have discovered personal and project pages by using generic pre-existing services (classifiers, named entity recognisers) when possible, and inducing a number of domain-specific recognisers in other cases. These recognisers are induced using only a short list of user-defined examples. Such recognisers are domain-specific, not site-specific, i.e. we just need one list of names in order to define a service that will work on all sites without any further adaptation. Finally we have found personal/project pages using different strategies such as discovering hyperlinks around known names and querying available search engines.

2.3 Extraction of personal data

The extraction of personal information from personal pages is generally quite easy. For example a generic IE system easily spots email addresses and telephone numbers, etc. and they tend to be unique in the page. For other information (e.g. the position of a person, such as "professor", "researcher", etc.) it is possible to use the procedure mentioned above for training domain-specific IE systems starting from a short list of examples.

2.4 Identifying communities of practice

For recognising the involvement of people in projects it is generally sufficient to extract all the people's names mentioned in a project page or its sub-pages; then it is necessary to relate such names to the list of site-specific known names, either using available hyper-links associated to the names (hyper-links are generally unique identifiers of people names) or to use more sophisticated methodologies such as those used for Natural Language Processing or even weaker Web-specific ones such as (Alani et al., 2002).

Identifying publications is much more a complex task. There are a number of publicly available services that provide publications (e.g. digital libraries such as CiteSeer) where co-authorship information is easy to extract because the output format is very regular and a wrapper is very easy to induce using some examples. Unfortunately such databases tend to be largely incomplete and sometimes a bit out of date. Most people/departments provide specific up-to-date publication lists. Identifying such pages is generally very easy because they contain a large number of citations of a specific person (in case of personal publication lists) or large number of staff names (in case of common pages) together with many occurrences of keywords such as "proceeding", "conference", "journal", etc., so they can be identified using a normal IR system. Unfortunately extracting co-authorship from such pages is quite difficult because the list format is generally page-specific and not easy to identify. We use the available digital libraries as a source for determining a preliminary list of papers' titles, co-authors and date of publication to be used for annotating such publication lists (as we did with the person and project names). Then we induce wrappers that will extract a more complete list. Again, we use both the redundancy on the Web (information located in multiple places such as in multiple publication pages and digital libraries) and the relatively rigid formatting of some (parts of) pages to learn more complete lists. In the case that such a strategy fails, it is possible to resort to less reliable methods, such as extracting all the people names and dates in their proximity (e.g. names and dates are in the same list item, so we assume they are related), but this is much less reliable and therefore to be used only as a last resort.

At this point we have all the information we need to extract communities of practice and their evolution in time.

3 Conclusion and Future Work

In this paper we have proposed a methodology for bootstrapped learning in order to extract information from Web sites, using very limited amount of user input. We have exemplified the methodology using a specific application, but the methodology is generic and can be safely extended to a number of other tasks by specifying different web resources. In the specific application, the only user input is a number of examples of the information to be extracted (e.g. project names lists). In other tasks, some limited manual annotation of examples could be the right way. What is important is that we have shown that the amount of user input can be dramatically reduced, when compared to fully supervised methodologies like (Vargas-Vera et al., 2002; Handschuh et al., 2002). The described methodology is applicable to cases where the information is likely to be highly redundant and where regularities in documents can be found. This is often the case of many repositories used for knowledge management and of Web pages belonging to specific communities (e,g, computer science Web sites, e-commerce sites, etc.). Other authors have shown that similar (but less sophisticated) methodologies can be successfully applied to retrieve very generic relations on the whole Web (Brin, 1998). Recent advances on wrapper induction systems show that the regularity required to induce wrappers is not as rigid as it used to be in the past. Current wrapper induction systems can very often be used on free texts (Freitag and Kushmerick, 2000; Ciravegna, 2001), making the methodology quite generic.

Qualitative analysis of results from preliminary experiments is satisfying. When Armadillo was run on a number of sites (such as nlp.shef.ac.uk and www.iam.ecs.soton.ac.uk), it managed to find most information using just a user-defined list of projects for **the first site**. We are currently performing other extensive experiments in order to test the accuracy of the data extracted from the CS web-sites, but thanks to the redundancy of the web, noise seems to be extremely low in our system. We have chosen this task because data on this topic is available from the University of Southampton (Shadbolt, 2002), so we will be able to compare our results with theirs (derived using semi-automatic methods). The comparison with such expensive human-based methodology will be a good evaluation of the added value of unsupervised methods to learning. Future work will involve the use of more sophisticated machine learning methodologies for unsupervised learning and the advanced use of data mining to discover new knowledge on the top of the current extraction methodologies in a way similar to that envisaged by (Ghani et al., 2000).

References

H. Alani, S. Dasmahapatra, N. Gibbins, H. Glaser, S. Harris, Y. Kalfoglou, K. O'Hara, and N. Shadbolt. 2002. Managing reference: Ensuring referential integrity of ontologies for the semantic web. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*.

Fabio Ciravegna. 2001. Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI)*. Seattle.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, Tampere, Finland.

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

Rayid Ghani, Rosie Jones, Dunja Mladenic, Kamal Nigam, and Sean Slattery. 2000. Data mining on symbolic knowledge extracted from the web. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000), Workshop on Text Mining*.

S. Handschuh, S. Staab, and F. Ciravegna. 2002. S-CREAM - Semi-automatic CREAtion of Metadata. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag.

N. Kushmerick, D. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997*.

Tom Mitchell. 2001. Extracting targeted data from the web. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, California.

Nigel Shadbolt. 2002. Caught up in the web. Invited talk at the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02".

M. Vargas-Vera, Enrico Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. 2002. MnM: Ontology driven semi-automatic or automatic support for semantic markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag.