Automatic Creation and Monitoring of Semantic Metadata in a Dynamic Knowledge Portal

Diana Maynard, Milena Yankova, Niraj Aswani, and Hamish Cunningham

Dept of Computer Science, University of Sheffield, Sheffield, UK
{diana,milena,niraj,hamish}@dcs.shef.ac.uk

Abstract. The h-TechSight Knowledge Management Portal enables support for knowledge intensive industries in monitoring information resources on the Web, as an important factor in business competitiveness. Users can be automatically notified when a change occurs in their domain of interest. As part of this knowledge management platform, we have developed an ontology-based information extraction system to identify instances of concepts relevant to the user's interests and to monitor them over time. The application has initially been implemented in the Employment domain, and is currently being extended to other areas in the Chemical Engineering field. The information extraction system has been evaluated over a test set of 38 documents and achieves 97% Precision and 92% Recall.

Keywords: semantic metadata, ontologies, dynamic knowledge management

1 Introduction

The h-TechSight project integrates a variety of next generation knowledge management (NGKM) technologies in order to observe information resources automatically on the internet and notify users about changes occurring in their domain of interest. In this paper we describe one part of the knowledge management portal, which aims at creating semantic metadata automatically from web-mined documents, and monitoring concepts and instances extracted over time. By tracking their usage and dynamics automatically, a user can be informed about new developments and other topics of interest in their field. We have developed a sample application in the employment domain, and are currently integrating other domains into the system.

1.1 Motivation

Employment is a general domain into which a great deal of effort in terms of knowledge management has been placed, because it is a generic domain that every company, organization and business unit has to come across. Many Human Resources departments have an eye open for knowledge management to monitor their environment in the best way. Many Recruitment Consultant companies

have watchdogs to monitor and alert them to changes. A number of job search engines (portals) have been launched using knowledge management extensively to link employees and employers[1] [2].

The employment domain contains many generic kinds of concepts. First this means that an existing Information Extraction system can more easily be adapted to this domain (because it does not require too many modifications), and second, it does not require a domain expert to understand the terms and concepts involved, so the system can easily be created by a developer without special domain skills. These two considerations are very important in the fast development of a system.

1.2 Knowledge Management Platform

The Knowledge Management Platform is a dynamic knowledge portal consisting of several different applications, which can be used in series or independently. These can be divided into two parts: tools for generic search (MASH) and tools for targeted search (ToolBox, WebQL and GATE). We shall concentrate here on the GATE tool.

GATE is used to enable the ontology-based semantic annotation of web mined documents. It is run as a web service which takes as input a URL and an ontology, and produces a set of annotations. The web service performs information extraction on the documents, and outputs an HTML page with instances of concepts highlighted. These results are stored in GATE's database and can be reused from another sub-module of GATE for statistical analysis.

2 Semantic Metadata Creation

There are several existing tools for semantic metadata creation, both semi-automatic and fully automatic.

Semi-automatic methods are generally more reliable, but require human intervention at some stage in the process. Usually this involves the user annotating data manually in order to provide training material for the system, which then takes over the annotation process. Examples of this kind of approach are MnM [10], S-CREAM[5] and AeroDAML[6]. These systems can usually be adapted to new domains and ontologies, but will need retraining by the user. This means that they are generally best suited to annotating large volumes of data within a single domain, and in situations where the user has an interest in investing some initial time and effort in the application. They are less suitable for the casual user who wants a ready-made tool to provide instant annotations for his data.

Automatic methods of annotation tend to be less reliable, but they can be suitable for large volumes of text where very high performance is not as paramount as having some kind of result. Because they require no human intervention, they are much more suitable for the casual user who wants a fast result,

[1] http://www.job-search.com/

[2] http://www.aspanet.org/solutionstemp/jobport.html

but does not want to invest time and effort in ensuring a very high quality output. Automatic methods tend to be more dynamic in that they can be adapted to new ontologies with no intervention. Ontology modification may also be a part of the process, thereby ensuring that a lifecycle is created by enabling feedback from the modified ontology to reflect in the application. Examples of automated tools are SemTag[4] and KIM[11]. Both of these systems find instances in the text using a large ontology, and perform disambiguation where instances are present in more than one place in the ontology. While SemTag aims more for accuracy of classification, KIM aims more for high recall.

3 Ontology-Based Information Extraction

3.1 GATE

GATE is an architecture for language-engineering developed at the University of Sheffield [2], which contains a suite of tools for language processing, and in particular, a vanilla Information Extraction (IE) system. In traditional IE applications, GATE is run over a corpus of texts to produce a set of annotated texts in XML format. In this case, however, the input to GATE takes the form of a set of URLs of target webpages, and an ontology of the domain. Its output comprises annotated instances of the concepts from the ontology. The ontology sets the domain structure and priorities with respect to relevant concepts with which the application is concerned.

GATE's IE system is rule-based, which means that unlike machine-learning based approaches, it requires no training data (see e.g. [9]). On the other hand, it requires a developer to manually create rules, so it is not totally dynamic. The architecture consists of a pipeline of processing resources which run in series. Many of these processing resources are language and domain-independent, so that they do not need to be adapted to new applications [8]. Pre-processing stages include word tokenisation, sentence splitting, and part-of-speech tagging, while the main processing is carried out by a gazetteer and a set of grammar rules. These generally need to be modified for each domain and application, though the extent to which they need to be modified depends on the complexity and generality of the domain. The gazetteer contains a set of lists which help identify instances in the text. Traditionally, this is a flat structure, but in an ontology-based information extraction (OBIE) application, these lists can be linked directly to an ontology, such that instances found in the text can then be related back to the ontology.

3.2 Employment ontology

For the employment domain in h-TechSight, a domain-specific OBIE application has been created, which searches for instances of concepts present in a sample Employment ontology. The ontology can be submitted as DAML+OIL or RDF, both of which are handled in GATE. The employment ontology has 9 Concepts:

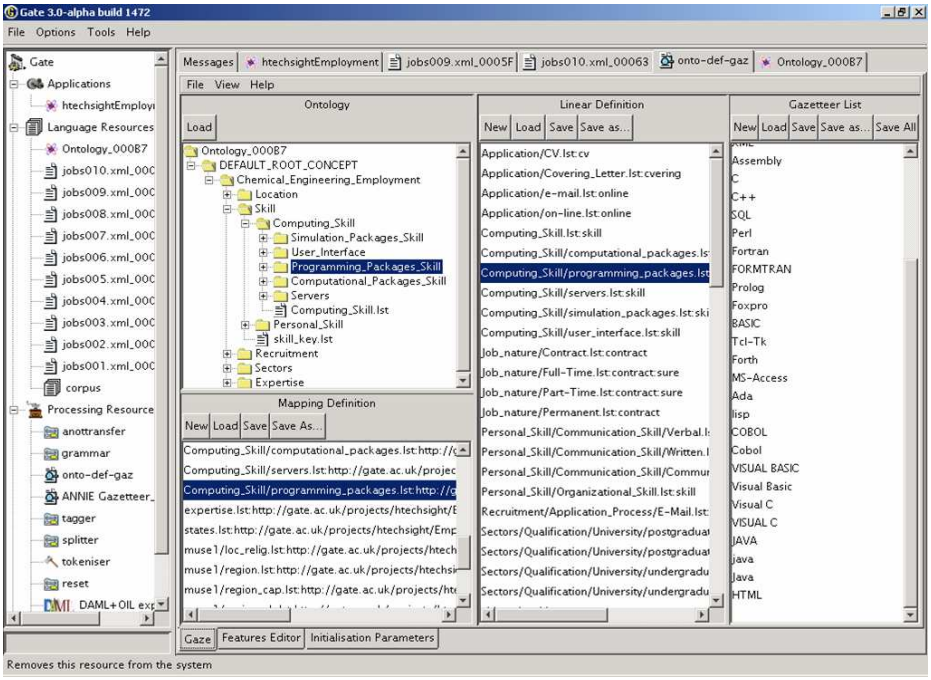

Fig. 1. Section of Employment Ontology in GATE

Location, Organisation, Sectors, JobTitle, Salary, Expertise, Person and Skill. Each concept in the ontology has a set of gazetteer lists associated with it. Some of these (default lists) are reused from previous applications, while others (domain-specific lists) need to be created from scratch. The default lists are quite large and contain common entities such as first names of persons, locations, abbreviations etc. Collection of these lists is done through corpus analysis (examining the texts manually and/or performing statistical analysis to spot important instances and concepts), unless a set of texts has been manually annotated by a user (in which case, the list collection process can be automatic [7]). Grammar rules for recognition of new types of entities mainly use these lists. However, there are also other lists collected for recognition purposes that contain keywords and are used to assist contextually-based rules. Some of the keyword lists are also attached to the ontology, because they clearly show the class to which the identified entity belongs. All lists that correspond to the ontology are ordered in a hierarchy similar to the class hierarchy in the ontology. A sample screenshot of a section of the ontology, the mappings from the lists to the ontology, and the contents of a list is shown in Figure 1.

The concepts (and corresponding instances) in which we are interested can be separated into 3 major groups. The first group consists of classic named entities

which are general kinds of concepts such as Person, Location , Organisation. The second group is more specific to the chosen domain of employment, and consists of the following types:

- JobId - shows the ID of posted job advertisements;
- Reference - shows the reference code of the job position;
- Status - shows the employment/position type;
- Application - shows the documents necessary and the method of job application (e.g. by email, letter, whether a CV should be sent, etc.);
- Salary - shows the information available in the text about salary rates, bonus packages, compensations, benefits etc.;
- Qualification - shows the qualifications required for the advertised position, mainly a University degree;
- Citizenship - shows restrictions about the applicant's citizenship and work allowance;
- Expertise - shows the required expertise / skills for the job;

For both groups, the grammar rules check if instances found in the text belong to a class in the ontology and if so, they link the recognised instance to that same class and add the following features:

```
EntityType.ontology = ontology url,
EntityType.class = class name
```

The third group presents instances already annotated with HTML or XML tags (if such exist), and consists of the following:

- Company - contains the name of the organisation advertising the job;
- Date_Posted - shows the date when the job advertisement was posted;
- Title - shows the job title;
- Sector - shows the sector of the job that is advertised;

If these are not already annotated in the texts, they are identified using further rules.

3.3 Grammar rules

The grammar rules for creating annotations are written in a language called JAPE (Java Annotations Pattern Language) [3].The rules are implemented in a set of finite-state transducers, each transducer usually containing rules of a different type, and are based on pattern-matching. In traditional IE applications, the rules find a pattern on the LHS, in the form of annotations, and on the RHS an action such as creating a new annotation for the pattern. In OBIE applications such as this, the rules also add information about the class and ontology on the RHS of the rule. So for example the string "PhD" found in the text might be annotated with the features:

```
{class = Postgraduate}
{ontology = http://gate.ac.uk/projects/htechsight/Employment}
```

This information is taken from the gazetteer, which is mapped to an ontology, as described earlier. The rules do not just match instances from the ontology with their occurrences in the text, but also find new instances in the text which do not exist in the ontology, through use of contextual patterns, part-of-speech tags, and other indicators.

In total the application contains 33 grammars, which run sequentially over the text. Each grammar contains anything from 1 to about 20 rules, depending on the complexity of the annotation type.

4 Export and presentation of results

The GATE application for the employment domain has been implemented in the h-TechSight portal as a web service. Here a user may input a URL and choose the concepts for the ontology. A new web page is created from the selected URL, with highlighted annotations. The result can be saved as an XML file.

Not only is the presentation of instances useful in itself, but furthermore, the occurrence of such instances over time is even more interesting. As well as the visual presentation, the results are also stored dynamically in a database and their statistical analysis is presented inside the hTechSight knowledge management portal. Results currently span January to June 2004. They have been collected by dynamically populating a Microsoft Access database with the following structure:

- Concepts: the concept which the record set of the database is about;
- Annotations: the instance of the record set annotated inside a document;
- Document_ID: a unique ID for the document;
- Time_Stamp: a time stamp found inside the document.

5 Monitoring instance-based dynamics

One of the most primitive dimensions of ontologies is the display of data as concrete representations of abstract concepts, i.e. as instances. Gate leads the data driven analysis in hTechSight, as it is responsible for extracting from the text instances represented in the ontology. These results are stored in a database and statistical analysis is invoked to present instance-based dynamics.

In the h-TechSight platform, we try to monitor the dynamics of ontologies using two approaches: dynamics of concepts and dynamics of instances. Users may not only annotate their own websites according to their ontology, but may also see the results of a dynamic analysis of the respective domain. They may see tabular results of statistical data about how many annotations each concept had in the previous months, as well as seeing the progress of each instance in previous time intervals (months). Following this analysis, end users may also

see the dynamics of instances with an elasticity metric that indicates the trend of each individual instance. Developments in the GATE results analysis have eliminated human intervention, as the results are created automatically in a dynamic way.

The two approaches to the monitoring of dynamics are described in more detail in the following sections.

5.1 Dynamics of Concepts

Dynamic metrics of concepts are calculated by counting the total occurrences of annotated instances over time intervals (per month). A visual representation of this analysis follows is shown in Table 1.

Concept	Count of instances for Jan	Count of instances for Feb
Application	4	46
Citizenship	0	9
Email	0	15
Expertise	0	1798
JobTitle	20	513
Location	0	13
Money	0	42
Organisation	0	420
Period	20	553
Qualification	4	51
Salary	12	200
Skills	74	1044

Table 1. Visualising Concept Dynamics

This analysis is dynamic in that counts of months are calculated automatically and new columns are added without human intervention. This automatic process is extremely useful, as a record of the performance of concepts is stored in a log file and may lead experts to useful conclusions and quick, wiser decisions. Occurrences per month may also help experts to monitor dynamics of specific concepts, groups of concepts or even the whole ontology. This analysis may help the decision making of stakeholders, by directing their resources according to the trends of the market of their domain.

5.2 Dynamics of Instances

By clicking on the concepts, a user may see the instances related to a concept. Instances are presented in a time series where the total occurrences per month and a calculation of an elasticity metric of instances are presented in tabular form. The elasticity metric (Dynamic Factor) counts the differences between

the total occurrences of every instance over time intervals (per month) taking into consideration the volume of data of each time period (documents annotated per month). Table 2 shows the dynamic factor (DF) and frequency of instances for the concept Organisation from January to March 2004. The higher the DF, the greater the upward trend. Looking at only 3 months of data does not give sufficient results for any conclusions to be drawn, but inferences can clearly be made from results over a longer period of time. Looking at the instances for the concept "Organisation" can monitor which companies are looking for new employees and track the recruitment trends for different companies. Monitoring instances for concepts such as Skills and Expertise can show which kinds of skills are becoming more or less in demand.

Instance	DF	Jan	Feb	Mar
ARC	145	-1	12	6
Archimedia SA	-1	0	1	0
Army	23	0	2	1
AT&T	-1	0	2	0
BA	23	0	3	1
BMI British Midland	-335	1	3	0

Table 2. Visualising the Dynamics of Instances

6 Evaluation

We conducted an initial evaluation of the IE application on a small set of 38 documents containing job advertisements in the Chemical Engineering domain, mined from the website `http://www.jobserve.com`. The web portal is mined dynamically using a web content agent written in a commercial web crawling software [1]. We manually annotated these documents with the concepts used in the application, and used the evaluation tools provided in GATE to compare the system results with the gold standard. Overall, the system achieved 97% Precision and 92% Recall. Table 3 shows the results obtained in more detail. The first column shows the annotation type. The next 4 columns show the numbers of correct, partially correct, missing and spurious annotations found. The last 3 columns show the figures for Precision, Recall and F-measure.

Some of the concepts show figure s of 100% Precision and Recall because they were taken directly from the original markup of the document (i.e. this information was already encoded in the HTML). The lowest performance was for concepts such as Skills and Expertise. This is unsurprising because this kind of information can be encoded in many different ways, and is hard to identify (not only for a rule-based system, but also for a learning-based one). We relied on contextual information and use of keywords to identify such concepts, but

Concept	Cor	Par	Miss	Spur	P	R	F
Person	7	1	1	0	93.75	83.34	88.24
Location	289	15	4	3	96.58	96.27	96.42
Organization	126	13	22	10	88.93	82.30	85.48
JobId	38	0	0	0	100	100	100
Reference	31	1	0	0	98.44	98.44	98.44
Status	42	1	0	0	98.84	98.84	98.84
Application	32	3	6	0	95.71	81.71	88.16
Salary	48	10	6	3	86.89	82.81	84.80
Qualification	57	15	9	5	83.77	79.63	81.65
Citizenship	19	2	0	0	95.24	95.24	95.24
Expertise	172	29	33	11	87.97	79.70	83.63
Skills	88	19	37	4	87.84	67.71	76.47
Willingness	4	0	0	0	100	100	100
Company	38	0	0	0	100	100	100
Date_Posted	38	0	0	0	100	100	100
Sector	38	0	0	0	100	100	100
Title	38	0	0	0	100	100	100

Table 3. Results of Annotating Sample Employment Corpus

the rules could be improved further. Overall, the results are very encouraging, however.

7 Conclusions

In this paper we have presented an application for automatic creation of semantic metadata from webpages in the Chemical Engineering domain. This is incorporated into a dynamic Knowledge Management Platform which also enables the monitoring of instances found and modification of the ontologies used. The application has been tested in the Employment sector with ecxcellent results, and is currently being ported to other genres of text such as news items and company reports. This involves adaptation of some of the rules, and integration of a new ontology more specific to the domain.

There are some problems still to resolved. For example, the system currently only adds class information to the instances found when the instances are already present in the ontology. It is still an open question how to decide where to link new instances, or where to link instances found in more than one place in the ontology. Currently we are investigating the use of coreference information and machine learning techniques to solve these problems.

References

1. Caesius. WebQL User's Guide: Introduction to WebQL. http://www.webql.com.

2. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
3. H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.
4. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of WWW'03*, 2003.
5. S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 358–372, Siguenza, Spain, 2002.
6. P. Kogut and W. Holmes. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *First International Conference on Knowledge Capture (K-CAP 2001), Workshop on Knowledge Markup and Semantic Annotation*, Victoria, B.C., 2001.
7. D. Maynard, K. Bontcheva, and H. Cunningham. Automatic Language-Independent Induction of Gazetteer Lists. In *Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004.
8. D. Maynard and H. Cunningham. Multilingual Adaptations of a Reusable Information Extraction Tool. In *Proceedings of the Demo Sessions of EACL'03*, Budapest, Hungary, 2003.
9. D. Maynard, V. Tablan, and H. Cunningham. NE recognition without training data on a language you don't speak. In *ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models*, Sapporo, Japan, 2003.
10. E. Motta, M. Vargas-Vera, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 379–391, Siguenza, Spain, 2002.
11. B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM – Semantic Annotation Platform. *Natural Language Engineering*, 2004. To appear.