Within the Framework of Course-assisted Creation,an Incremental Method to Extract Relevant Information from the Web and Integrate it in a Course Draft.

Pierre POMPIDOR, Michel SALA, Danièle HERIN

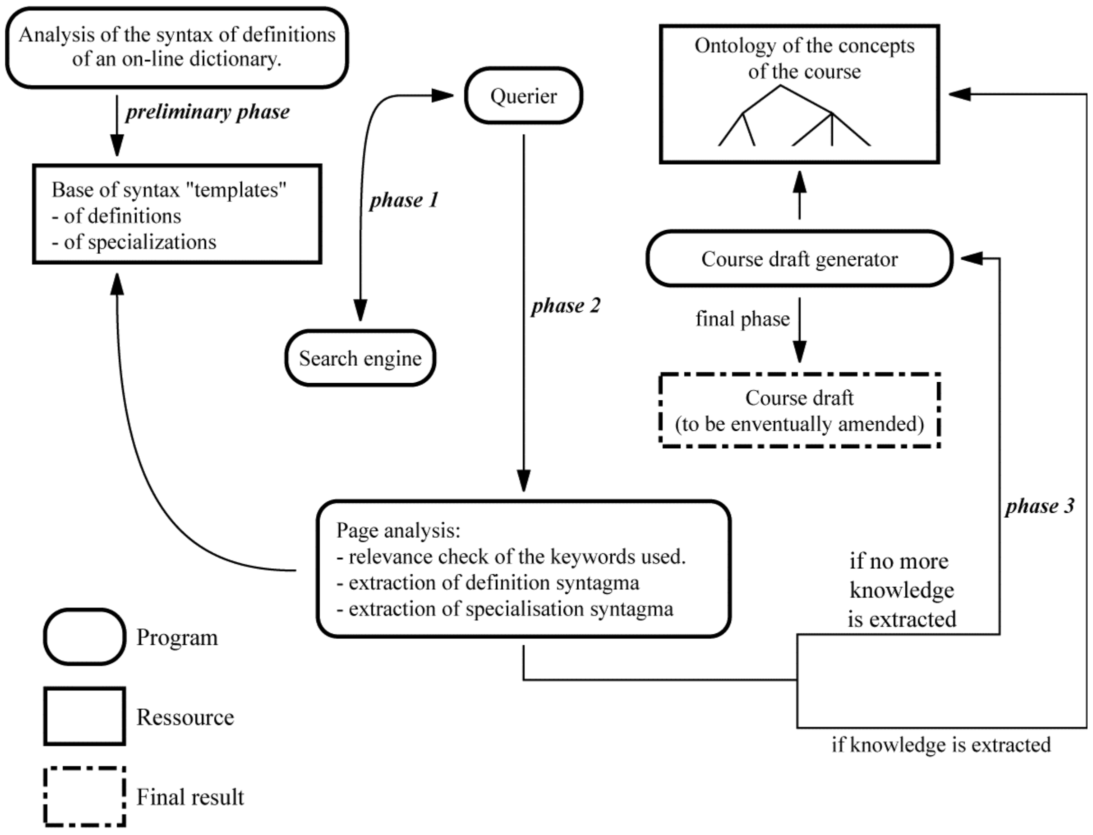
LIRMM
161 rue Ada, 34090 Montpellier, France
pompidor, sala, dh [@lirmm.fr]

Abstract. Within the framework of course-assisted creation, we are developing a prototype whose goal will be to enrich an ontology of teaching concepts. To do that, we automatically query search engines with key words extracted from pages that were previously analysed (beginning with an ontology which treats on a hierarchical basis the first keywords). This analysis (lexical, syntactic and semantic), targeted towards the extraction of definitions ("this concept-X is a ...etc.") and of specializations ("this concept-X comes in ...etc."), is based on the exploitation of a base of syntactic templates automatically learned from the analysis of definitions of several on-line dictionaries, and from a minimal dictionary (in particular to treat synonymies). The ontology thus created finally generates a course draft that must be "finalized" manually, in spite of the use of a few subroutines for knowledge synthesis (for the moment our work is mainly focused on the enrichment of the ontology). We illustrate this method with the creation of a course on "multi-tier architectures" for which we suppose only the bases initially known.

Introduction

Although the methods for indexing Web pages have considerably improved in recent years, (As shown by the effectiveness of search engines Google and Teoma, resp. [2] and [3]), the relevance of the answers provided is unfortunately still far from meeting internet users' expectations and, in particular, teachers trying to find content for their courses.
While we were working on a project of synthesis and semi-automatic aggregation of knowledge extracted from the Web (and not annotated beforehand for the purpose of being used to create new electronic documents [4] [6]) to assist the creation of on-line courses, two problems arose quickly: the imprecision of the requests that we tried to formulate, and the inaccuracy of the answers provided by the search engines. Indeed, we need to find precise and relevant information, with requests that must be correctly formulated, using adequate Web resources, and not be pointed towards a page or worse still, a too-general gateway. It is with this goal that we have developed a prototype that automatically queries one or more search engines, using increasingly elabo-

rate keywords. These key words will be integrated incrementally in an ontology [5] which is not only used to question the engine in an increasingly precise way, but which also represents the framework of the course being made. These key words will be extracted incrementally from the pages analysed, except for the very first ones that must be manually integrated in the initial "starter" ontology. The lexical/syntax/semantic analysis carried out to extract these key words is done using a base of syntactic templates extracted from the analysis of several thousand definitions of various on-line dictionaries, and applies only to the patterns of definitions or specializations. A final draft of the course is generated when no new key words are being integrated in the ontology. For the moment, this course is just a set of definitions and encyclopedic material.



Analysis of the syntax of definitions of an on-line dictionary.
preliminary phase
Base of syntax "templates"
- of definitions
- of specializations
Querier
phase 1
Search engine
phase 2
Ontology of the concepts of the course
Course draft generator
final phase
Course draft (to be enventually amended)
phase 3
Page analysis:
- relevance check of the keywords used.
- extraction of definition syntagma
- extraction of specialisation syntagma
if no more knowledge is extracted
if knowledge is extracted
Program
Ressource
Final result

Figure 1. The general operation of our prototype

1. The main methods for indexing used by search engines and their shortcomings

First of all, let's examine the methods used by the search engines to measure the relevance of a Web site (or more precisely of a Web page). These methods determine the order of appearance in the list of results.

- Manual referencing of the main web servers of a particular domain. This method for directory construction is currently reaching it's peak with DMOZ (Open directory Project) [1] which references over 1,400,000 sites spread across 200,000 categories.
- Link popularity, the number of pages linked to a site, is used by many search engines to select pages for indexing (HITS and PageRank methods [7]).
- Memorisation of the pages that were selected by the web user in the list of results. Ideally the engine will detect not only the documents that were chosen by the user but also most particularly those that were looked at for a long period. This method is not without drawbacks (returned lists must be able to send back information to the server). The DirectHit company absorbed by Teoma [3] proposes methods based on this type of measurement.
- Asking users to vote for their favourite web sites. This method is obviously too skewed and subjective.

Unfortunately, in practice these methods are insufficient for our needs. Thus, the most interesting resources for a practical computer course (for example: multi-tier architectures) should be:

- on-line courses (beware of plagiarism!)
- official web sites of the various technologies covered in the courses (for this example : servlets, ASP, PHP, EJB, Corba, DCOM, ...)
- discussion forums (FAQ...)

However, if we query Google or another search engine such as Teoma (for which the results, in this case, are better) with "multi-tier architectures", these methods of indexing will only refer a few interesting pages lost among a large amount of service offers from companies, resumes, titles of articles and conferences. And still, official sites of the technologies covered in the course will appear among the few interesting URLs generated. These will lack depth or be dangerously skewed (a web site on servlets will not mention PHP or will mention it in a negative way, and reciprocally). In any case these sites have a strong encyclopaedic organization that does not favour the creation of a course, while discussion forums often discuss minor details (often extremely interesting, but too specific for use in a course). The exploitation of courses already on line is thus kept back by queries that are too low in relevant key words (which is trivial during the exploratory development of a course). Our objective is thus to select keywords (from the already analysed pages) that are progressively more and more relevant to the topic of the course.
Let us illustrate our method on a concrete example (the examples provided are translated from French (we use French syntactic templates) but real).

2. The example

Let us take as an example the following problem: we want to semi-automatically create a course on multi-tier architectures for which we know only a few concepts. Initially, we create an ontology that treats on a hierarchical basis the three main keywords (and their synonyms) that we know:

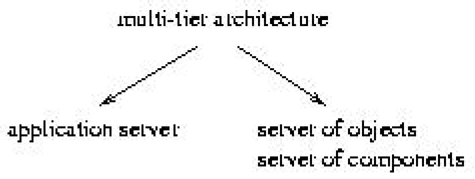


Figure 2. This initial ontology will be used by the querier

3. The querier (written in Perl)

The querier will query Google (or one or several other search engines) with a list of keywords taken from each branch of the ontology. That is to say, in our example, with one of the following combinations:

- « multi-tier architecture » « application server »
- « multi-tier architecture » « object server »
- « multi-tier architecture » « server of components »

(Composite keywords have to be found in one block (which implies the use of quotation marks), but can be inflected ("multi-tier architectures"...)
It then imports the referenced pages and transmits them to the analyser.

4. The analyser (written in C)

The analysis being carried out is based on the light syntactic analysis which we developed for an automated system of service identification on Web pages named "Chimère" [8]. For each concept of the ontology, the analysis consists of extracting two types of knowledge:

- the definitions which give an explanation of this concept
- specializations which list instances of this concept (often by means of enumerations)

This analysis is based on syntactic templates automatically learned from the analysis of thousands of definitions of several on-line dictionaries. The learning carried out is based on frequency calculations on the under-trees of the syntactic trees generated by syntactic analysis of the field labels of these dictionaries. For example, this learning revealed that a certain number of definitions followed the following template (it was the first learned) : Concept-X : the act of [participle present]. The patterns inside the brackets represent either syntactic phrases (leading capital letter) or lemmas (leading lower-case letter). The position of the concept is represented by "Concept-X" (or its referent: "referent-of-Concept-X"), and by "generalization-of-Concept-X " if it is generalized.

Here are the first three examples of the first example template:
acceleration: the act of accelerating
acceptation: the act of accepting
accumulation: the act of accumulating or amassing ...

Here are a few examples of definition templates used in our examples:
[Definite article] Concept-X [to be] [Definite article] ...
[Definite article] ... Concept-X [to separate] ...
[to be] a generalization-of-Concept-X in which ...

And some examples of specialization templates also used:
Concept-X [to be] [Enumeration]
[Definite article] Concept-X [to be composed of] ..
[to be] [Definite article] Concept-X

This analysis is multifaceted:

- lexical as it allows lemmatisation of words present in the pages (determination of their canonical form from their flexional form)
- syntactic as it allows pairing of base templates.
- the pairing of surface templates is an old and tested cursory technique. Although it only enables us to analyse about two thirds of the potentially relevant sentences present in the documents referenced by the search engines, it is extremely effective speed-wise. We are using an expert system to do this.
- and semantic, by determining the referents (primarily pronouns representing the desired concept). Several examples will be given further in this article.

5. Extraction of definitions

Here are three examples of knowledge extraction done on pages returned by search engine Google using the combinations of keywords listed previously. The matching keywords are boldfaced and the extracted definitions are underlined.
First example:

An **application server** is based on a **multi-tier architecture**. It is a model of architecture of applications in which one separates the presentation, data processing and data. The aim is to allow an evolution of the one of these three tiers in a way relatively independent of two others.
From: http://users.skynet.be/johant/servapp.htm

- The keywords "application server" and "multi-tier architecture" match
  - ➔ the page is analysed
- The template "[to be] a generalization-of-Concept-X in which" matches
  - ➔ " in which one separates the presentation, data processing and data " is extracted
- The last sentence cannot be analysed

Second example:

> The **application server** is <u>the environment of execution of the server-side applications</u>. **It** <u>deals with the sets of the functionalities which make it possible to N clients to use the same application</u> : ... .The **application server** is thus essential if one wishes to avoid re-developing the whole of these functionalities (case of cgi). For this reason, the JSP/Servlets engines, Microsoft ASP, Cold Fusion, PHP ... are **application servers** (even if they are integrated into the Web server PHP/ASP).
> From: http://medias.obs-mip.fr/cours/sgbd/cours013.html

- The keyword "application server" matches (thrice) -> the page is analysed
- The template [Definite article] Concept-X [to be] [Definite article] matches
  - ➔ "the environment of execution of the side-server applications" is extracted.
- The referent "It" is defined (it is validated by application server)
- The template referent-de-Concept-X matches
  - ➔ "deals with the sets of the functionalities which make it possible to N clients to use the same application" is extracted.
- The last sentence will be analysed by a specialisation template.

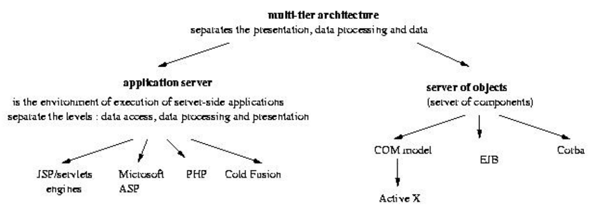
Third example:

> The new **application servers** <u>separate the levels: data access, data processing and presentation</u>. **They** work starting from reusable objects components EJB ou COM/DCOM. These components can be distributed and accessible through a directory. These components can ensure, in a more or less hidden way, the functions of transactional monitor, the persistence of data, the load balancing.
> From : http://www.eikonex.org/article.php3?id_article=11

- The keyword "application server" matches -> The page is analysed
- The template [Definite article] ... Concept-X [to separate] ... matches
  - ➔ « separate the levels : data access, data processing and presentation » is analysed
- The referent "They" is defined -> but it does not lead to immediate analysis

6. Extraction of specialisations (in particular via enumerations)

The matching keywords are boldfaced and the extracted definitions are underlined.
First example:


multi-tier architecture
separates the presentation, data processing and data
application server
is the environment of execution of server-side applications
separate the levels : data access, data processing and presentation
server of objects
(server of components)
JSP/servlets engines
Microsoft ASP
PHP
Cold Fusion
COM model
EJB
Corba
Active X

The **application server** is the environment of execution of the server-side applications. It deals with the sets of the functionalities which make it possible to clients to use the same application : ... .The **application server** is thus essential if one wishes to avoid re-developing the whole of these functionalities (case of cgi). For this reason, the JSP/Servlets engines, Microsoft ASP, Cold Fusion, PHP ... are **application servers** (even if they are integrated into the Web server PHP/ASP).
From: http://medias.obs-mip.fr/cours/sgbd/cours013.html

- The keyword matches were already explained.
- The template "... [to be] [Definite article] Concept-X" matches
- ➔ extraction of "the JSP/Servlets engines, Microsoft ASP, Cold Fusion, PHP ..."

Second example:

To manage these objects, an environment of exploitation is necessary : the **object server**. This **object servers** will have to provide services completely different from those of the application servers. In short, one has to make with techniques very differents. The principal **object servers** are EJB servers (Enterprise java Beans), Corba, microsoft DCOM. **They** are not necessary to these developments only if one wishes to use the logic of business objects fully.
From : http://medias.obs-mip.fr/cours/sgbd/cours013.html

- ➔ The keyword “object server” matches (thrice) -> The page is analysed
- ➔ The template "Concept-X [to be] [Enumeration] ..." matches
- ➔ extraction of " EJB servers (Enterprise Java Beans), Corba, ... DCOM "
- ➔ The referent “They” is defined
- ➔ ->but it does not lead to immediate analysis (no matching template)

Figure 3. From these extractions an initial enrichment of the ontology is carried out

7. Annotation of the new concepts

A new analysis is made to annotate the newest concepts integrated in the ontology that is to say in our example: JSP/servlets engine, Microsoft ASP, PHP, Cold Fusion, COM, ActiveX model, EJB, Corba.
To do that, the analysis will recover the already analysed pages to associate additional information to these new key words (which we refer to as annotations because we no

longer use the preset templates, for more flexibility). Generally speaking, this analysis is reapplied on a page as long as new annotations are being extracted.
The new keywords are boldfaced and the annotations are underlined.

The new application servers separate the levels : data access, data processing and presentation. They work starting from reusable objects components **EJB** or **COM/DCOM**. These components can be distributed and accessible through a directory. These components can ensure in a more or less hidden way, the functions of transactional monitor, the persistence of data, the load balancing, ...

➔ "EJB" et "COM" are annotated as being "reusable objects components"
➔ This allows us to carry out another analysis on the same text.

The new application servers separate the levels : data access, data processing and presentation. They work starting from reusable objects components EJB or COM/DCOM. These **components** can be distributed and accessible through a directory. These **components** can ensure in a more or less hidden way, the functions of transactional monitor, the persistence of data, the load balancing, ...

Two additional annotations are extracted:
➔ for "EJB" and for "COM": "can be distributed and accessible through a directory"
➔ for "EJB" and for "COM": «can ensure in a more or less hidden way, ...»

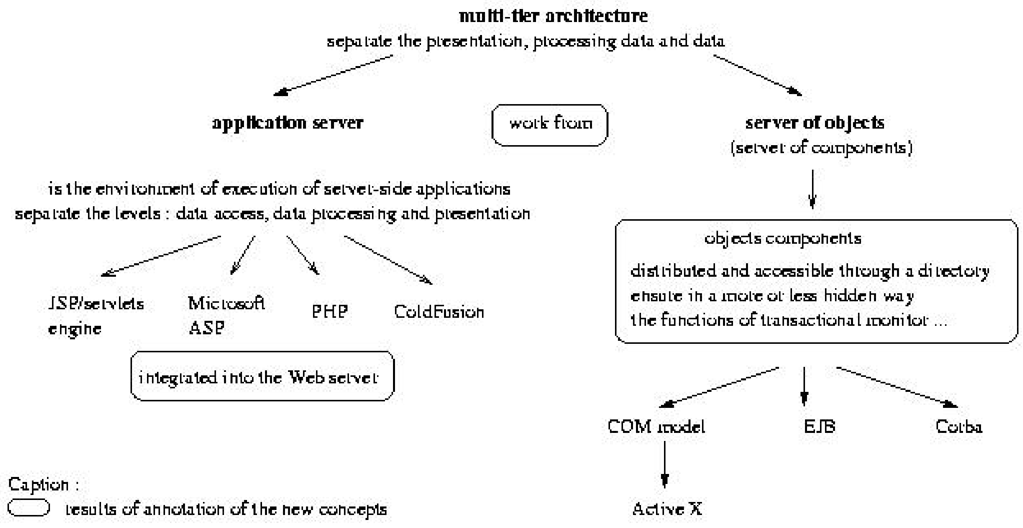

multi-tier architecture
separate the presentation, processing data and data
application server
work from
server of objects
(server of components)
is the environment of execution of server-side applications
separate the levels : data access, data processing and presentation
JSP/servlets engine
Microsoft ASP
PHP
ColdFusion
integrated into the Web server
objects components
distributed and accessible through a directory
ensure in a more or less hidden way
the functions of transactional monitor ...
COM model
EJB
Corba
Active X
Caption :
results of annotation of the new concepts

Figure 3. A new enrichment of ontology is carried out

From this enriched ontology, the querier transmits new requests to the search engine. The rules used to construct those requests are somewhat different to those applied to the initial ontology: (i)the keywords/Concepts are always grouped according to the

branch of the tree (from the root of the ontology to one of its leaves) ; (ii)but the keywords/Concepts are no longer linked (no more quotation marks)
These construction rules will use Google's selection strategies, which will try to satisfy all the keywords (Google automatically makes them obligatory). For example a first query on “multi-tier architecture” returns 9090 answers, 1930 answers if the keyword “application server” is added, 25 if the keyword “JSP engine” is added. It will ensure the number of documents to be finally analysed will be restricted.
In our example, here is a list of keywords that make up the requests: (i)from : multi-tier architecture application server JSP/servlets engines; (ii)multi-tier architecture application server Microsoft ASP; (iii)muli-tier architecture application server PHP
(iv) ... ; (v) multi-tier architecture object server components COM model ActiveX, (vi) multi-tier architecture object server components EJB (vii) to : multi-tier architecture object server components Corba ; (viii) Which enables us, for example, to reach others documents which will again enrich the ontology...

8. The course scripts generator (written in Perl)

As our method converges, the ontology finally stops evolving and the course draft generator produces a synthetic text (frequently malformed) for which the concepts of the ontology are the framework.

In our example, the generated course draft is:

A <u>multi-tier architecture</u> separate presentation, processing data ans data. A multi-tier architecture is composed of application server and object server.

An <u>application server</u> (JSP/servlet engine, Microsoft ASP, PHP et Cold Fusion (Microsoft ASP et PHP integrated into the server Web)) is the environnement of execution of server-side applications, and separate the levels: data access, data processing and presentation.

An <u>object server</u> manage objects components distributed and accessible through a directory. The object composents ensure, in a more or less hidden way, the functions of transational monitor.

This script obviously needs amending. For example, the last generated sentence is rather contemptible.

9. Conclusion

Our method, and the associated prototype, are beginning to produce concrete results. However, this procedure implies on the one hand a strong assumption, and is limited on the other hand, by a certain number of constraints and shortcomings.
Our final assumption is that we are building a course using resources for which no scientific or pedagogic guarantee is given, other than their appearance at the top of the results list of recognized search engines (tests on Google and Teoma). Indeed, these documents are neither referenced, nor annotated semantically beforehand to

contribute to the creation of new electronic documents as in [4] [6]. Ideally, we should be able to compare the generated course draft with an online course of reference in the field.
In addition, the constraints and the shortcomings are as follows:

- ➔ The initial ontology, even minimal, must precisely define the problematics (or risk building a forest of trees), and must be correct (the key words must obviously be coherent and non-redundant).
- ➔ The techniques for knowledge synthesis that we apply to it are for the moment too basic to avoid the duplication of definitions, or superfluous annotations.
- ➔ And most importantly, we have not yet integrated revision mechanisms which would make possible the detection and elimination of contradictory definitions.
- ➔ This last point is undoubtedly the most interesting and important to develop.

References

[1] http://dmoz.org/
[2] http://www.google.com/technology/
[3] http://www.teoma.com/
[4] Al-Tawki Y., "SABRE, an authoring system based on reuse of documents"
[5] Gruber T., "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition 5, p. 19-220, 1993
[6] Levy D. M., "Document reuse and document systems", Electronic publishing, vol. 6(4), p. 339-348, 1993
[7] Page, L., Brin, S., Motwani, R., Winograd, T., « The PageRank Citation Ranking: Bringing Order to the Web », Computer Science Department, Stanford University, , TechReport, 1998
[8] Segret M.-S., Pompidor P., Hérin, Sala M. , "Utilisation d'ontologies pour intégrer des informations semi-structurées issues de pages Web", INFORSID, Mai 2000