Automated Metadata Hierarchy Derivation

Amjad ABOU ASSALI   Hugo ZANGHI

Université de Technologie de Compiègne
Compiègne, FRANCE

amjad.naa@gmail.com   *hugo.zanghi@gmail.com*

Abstract

This paper presents an automated approach for building a metadata hierarchy of a set of web sites without the use of any predefined external hierarchies, and then merging and comparing them. The nodes of the hierarchy are the keywords of the specified web sites, and the links between these keywords are the weak subsumption relationships. We apply this method in the RTGI[1] project [8] on clusters of web sites already defined. The hierarchies can show how homogeneous each cluster is and permit to outline the contents of each corresponding cluster effectively. Moreover, we construct the common hierarchy of multiple clusters so that we check if their individual hierarchies are well distinguished and separated in the common one, which in turn indicates the correctness of clustering. At the end, we build the Semantic-hypertext graph of the sites which explains the semantic contents along with the topological structure of the sites.

Keywords— *Metadata hierarchy, Subsumption, Semantic-Hypertextual graph, Part-Of-Speach tagging, Subject/verb dependencies, Synonyms.*

1 Introduction

Hierarchies are an efficient structure to describe and understand information, and they can be used to browse and summarize large sets of documents. Therefore, finding methods to build these hierarchies was the goal of many researchers whose most methods were manual whereas some others were automated. We all agree that building a metadata hierarchy manually gives better results, but in exchange, it needs much more human efforts which tends to be difficult to afford sometimes. Thus, we are interested, in this paper, to develop an automated approach to deal with this problem and that depends on the subsumption relationships.

This approach is a part of the RTGI project where:

- Thousands of web sites of a specific domain are collected;
- Their contents are indexed;
- The *topological* relationships (i.e. hypertext links) between these sites are analyzed (by means of the *Hubs & Authorities* of Kleinberg) so that we get topological clusters of web sites;
- Finally, these clusters are visualized in a graph of nodes (*web sites*) and arcs (*hypertext links*).

Here intervenes our method to build a content-oriented hierarchy of a cluster of web sites so that the important keywords are shown in a graph where the most related keywords are close to each other; In addition, we construct the graph of the common metadata hierarchy of multiple clusters to check if these clusters are really different from each other.

An essential and original goal of this work is to verify if the topological clusters of web sites are also semantic ones. Thus, we present how to project a metadata hierarchy on topological clusters; i.e. how to make the fusion between the hypertextual and the semantic graph.

[1] Réseuax, Territoires et Géographie de l'Information

We begin by some achieved work in this domain (section 2), then we explain the steps to reach our hierarchy (section 3) and we propose how to enrich this hierarchy (section 4) by gathering the synonymous words into one node in the hierarchy. At the end, we present our experiments and results (section 5), and we show how we can benefit from our results to make a great fusion between the topological graph of web sites and their metadata hierarchy to produce the semantic-hypertextual graph (section 5.1).

2 Related work

The problem of constructing a metadata hierarchy was the interest of many researchers who followed different methods to deal with it, and only few of which were automated. Stoica and Hearst [4] obtain their hierarchy depending on the already defined hypernym paths of the WordNet thesaurus. Rydin [5] uses the lexico-syntactic constructions as indicators of the hypernym relations between words; for instance, in a phrase such as:

$$\textit{such } NP_h \textit{ as } ((NP_2) * \ NP_n \textit{ and}|\textit{or}) \ NP_1$$

the noun phrase NP_h is a hypernym and the other noun phrases $(NP_i)_{i\in(1..n)}$ are its hyponyms. In [3], tools such as TreeTagger and LoPar are used to finally get a list of verb/(subject, object, and prepositional phrase) dependencies from which they form the attributes of each word and then apply the FCA (Formal Concept Analysis) method to build the lattice of the words and then the hierarchy. Sanderson and Croft [1] use the notion of *subsumption* that they define as follows; for two terms, x and y; x is said to subsume y *iff*: $P(x/y) \geq 0.8$ and $P(y/x) < P(x/y)$; i.e. if the documents in which y occurs are approximately a subset of those in which x occurs. Therefore, x is considered the parent of y in the resulted hierarchy.

The former method attracted us because when x subsumes y in a set of documents, this means that x is more general than y and consequently x is more representative for the documents than y.

3 Building the hierarchy

In order to build a metadata hierarchy for a set of web sites, we must first pick the right representative words out of these sites. These words are chosen to be just the nouns. Then, we specify which pairs of these words constitute a subsumption relationship, and build their hierarchy. We benefit from this hierarchy by designing the *semantic-hypertextual* graph of the sites whose nodes are the sites with their general keywords and whose arcs are the topological (site $\rightsquigarrow$ site) and semantic (site $\rightsquigarrow$ keyword) links. We visualize these graphs using tools such as Pajek and GUESS[2] which allow us to analyze the results obtained (Figure 1 shows the different steps of our approach).

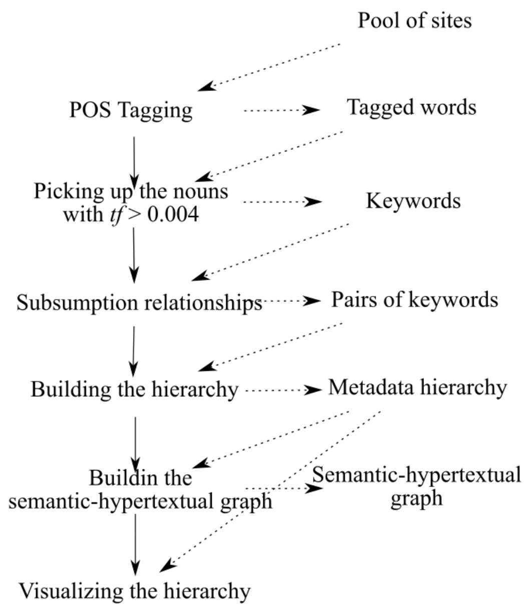

Figure 1: Main steps of the method

3.1 Part-of-speech tagging

In our method, we're only interested in extracting the nouns as keywords out of the web sites. Thus, we need to know the part-of-speech tag (i.e. noun, verb, adjective, etc.) of each word. To be able to do this, we make use of the program "TreeTagger"whose job is to parse an input text file to specify the POS tags of every word in it along with its root (infinitive) (e.g. vote $\rightsquigarrow$ voter) (*c.f.* Figure 2). Thus, for each web site, we passe its textual contents to the tagger to get the tag of each word and then keep only the nouns.

At the same time, we compute the frequencies (number of occurrences) of every noun in the site. To choose the nouns that most represent their sites, i.e. the keywords, we first filter

[2]Graph exploration systems

Cette	PRO:DEM	ce
situation	NOM	situation
met	VER:pres	mettre
en	PRP	en
évidence	NOM	évidence
le	DET:ART	le
caractère	NOM	caractère
incontournable	ADJ	incontournable
de	PRP	de
la	DET:ART	le
place	NOM	place
du	PRP:det	du
travail	NOM	travail
dans	PRP	dans
un	DET:ART	un
pays	NOM	pays
industriel	ADJ	industriel
.	SENT	.

Figure 2: Part of a tagged file by TreeTagger

all the nouns using a predefined stop list, then, having the frequency of each noun, we calculate the normalized term frequency (*tf*) values of each noun by the function:

$$tf_i = \frac{f_i}{\sum_{j=1}^{n}(f_j)} \quad (1)$$

where tf_i is the term frequency of the word i, f_i is its frequency, and n is the number of the whole nouns in the site.
Finally, we take only those with *tf* values greater than an arbitrary value (0.004) that was chosen by experiment, and obtain consequently our set of *keywords*.

3.2 Subsumption relationships

The hierarchy that we are going to build contains one type of relations between its nodes; the *subsumption* relation, which is originally defined as: For two keywords x and y; we say that x subsumes y *iff*:

$$P(x/y) = 1 \;\; and \;\; P(y/x) < 1 \quad (2)$$

where $P(x/y)$ is the conditional probability of x given y.
These two conditions(2) imply that the documents in which y occurs are a subset of those in which x occurs. Consequently, x is the parent of y in the hierarchy. As mentioned in [1], the previous two conditions are very strict, and many suitable pairs (x, y) fail because a few occurrences of the term y don't co-occur with x. To avoid this, these two conditions were relaxed and redefined to get the notion of the *weak* subsumption as follows:

For two keywords x and y; we say that x weakly subsumes y *iff*:

$$P(x/y) \geq 0.8 \;\; and \;\; P(y/x) < P(x/y) \quad (3)$$

With these two conditions in mind, we scan all the chosen keywords and compute the conditional probabilities for every pair of them as defined by the two following equations:

Let W be the set of keywords; we have:

$$\forall x, y \in W; \; P(x/y) = \frac{N_{x,y}}{N_y} \quad (4)$$

$$\forall x, y \in W; \; P(y/x) = \frac{N_{x,y}}{N_x} \quad (5)$$

where $N_{x,y}$ is the number of documents containing the two words (x,y) together; and N_y and N_x are the number of documents containing the words y and x respectively.
We take into consideration only the pairs (x,y) that achieve the conditions(3), and in the case where $P(x/y) = P(y/x)$, we choose one of the two words arbitrarily to be the parent of the other. If the two terms co-occur together in two or fewer web sites, we don't consider their subsumption relation [2].

Because of the transitive nature of subsumption, we tried to eliminate the arcs in the hierarchy in the manner: If x subsumes y, y subsumes z, and x subsumes z, then we eliminate the relation (x,z). But we noticed that the complete hierarchy with all of the relations better explains the contents of the web sites than the reduced one, so we kept working with the full hierarchy.

At the end of this step, we get the metadata hierarchy that we look for and it's the time to visualize it and make sure if it was really representing the contents of its web sites.

3.3 Visualizing the hierarchy

To visual the hierarchy, we use special graph exploration systems, "Pajek"[3] *&* "GUESS"[4], that provide many algorithms and techniques to manipulate an input graph. These two systems allow to draw a graph of nodes and arcs where the distance between the nodes depend on the links between them; i.e. the more the nodes are linked, the more they get close to each other. This property is very useful to our hierarchy since we are interested to show which terms are more related to each other according to the subsumption relationships between them.

[3]http://vlado.fmf.uni-lj.si/pub/networks/pajek/
[4]http://graphexploration.cond.org/

4 Grouping the synonyms

Let's now go further and try to enrich the constructed hierarchy. We search for the synonymous words in the set nouns and consider them as one concept in the hierarchy. In this way, the number of nodes may be reduced, and some interesting implicit relationships can be found.

4.1 subject/verb dependency relations

It is very important to reduce the number of keywords in an efficient way so that we don't lose information and also give rise more precise content-oriented information. To do this, we put each group of synonymous words into one concept in the hierarchy.

We know that each word may have several synonyms depending on the local context of the word. Thus, in order to find the right synonyms of a word, we depend on the following assumption: *"Two different words are likely to have similar meanings if they occur in identical local contexts"*[9]. These local contexts are considered, in our approach, the dependency relations "subject/verb".

To achieve this goal, we use the program "INTEX" that takes as an input a text file and parses it to identify the sequences of words that correspond to a specified pre-built Finite State Transducer[5] (FST). Thus, we had to build the transducer that allows to extract the subject/verb dependency relations out of a text.

In fact, the sequences recognized by INTEX may be sometimes erroneous because of the faults it makes when deciding the POS tag of the word. So, in order to know the correct resulted sequences, we use the previous results of the TreeTagger, which are more accurate, and compare them with the results of INTEX. Finally, for each keyword, we keep a list of the verbs with which it comes in the sites.

4.2 Synonyms grouping

Now that we have each keyword with its list of verbs, we search for its candidate synonyms in the famous French synonyms' dictionary "Le Bailly", and we only keep those candidates that appear in the sites. Then for each keyword, we measure the vectorial similarity between its vector of verbs and the vectors of its candidates according to the relation:

$$S(x,y) = \frac{\sum_{v\in V}(v_x v_y)}{\sqrt{\sum_{v\in V} v_x^2 \sum_{v\in V} v_y^2}} \qquad (6)$$

Where v_x and v_y is the number of times where x and y comes with the verb v, respectively. Then we say that x and y are synonyms if they have great similarity value towards each other in comparison with those similarity values with the other candidate synonyms.

5 Experiments and results

We applied this method on four different clusters built by experts of the RTGI group, and in every one of these clusters there were about 8 sites. The theme of these clusters are already known.

First, we found the metadata hierarchy of each cluster alone according to the method explained previously, and we visualized them using Pajek and GUESS. The resulted hierarchies were logically coherent.

The first cluster talks about the village (la ville), or in other words, the life of a municipality (*c.f.* Figure 3). Some of the interesting keywords that we found were: (*ville, maire, association, conseil*) and they were in the center of our graph, which means that they are the main representative words of this cluster and this is right. On the other hand, we see on the sides the arguments of action such as (*logement, emploie, cultur*).

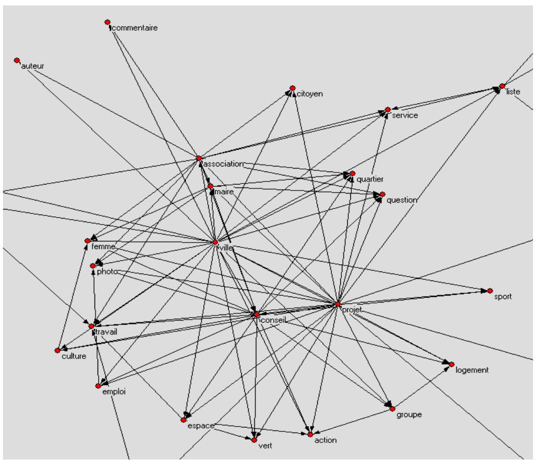

Figure 3: Part of the metadata hierarchy of the first cluster

We found also a very interesting relationship between *'espace'* and *'vert'* which tells us that the

[5] A paradigm of Deterministic Finite-state Automata (DFAs)

keyword *'vert'* belongs to the term *'espace vert'* and not to the term *'parti verte'*.

By visualizing the metadata hierarchy of the second cluster which talks about Europe (*c.f.* Figure 4), we see in the middle of the graph the keywords: (*projet, conseil, constitution, droit, citoyen*), and on the sides we see also more specific keywords such as *'exigence'* and *'socialiste'*, which is a very reasonable result.

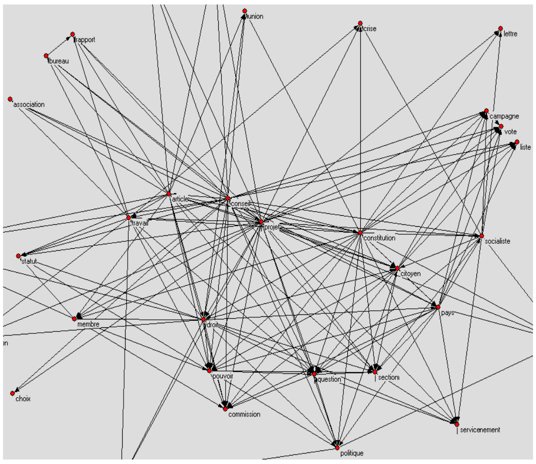

Figure 4: Part of the metadata hierarchy of the second cluster

By the same way, we construct the metadata hierarchy of the two other clusters that talk about the "Yes of the social party"(Le *oui* du partie socialiste) and the "No of the extreme party"(Le *non* d'extrême gauche) in France against the European constitution, and we found satisfying results.

Now, it's the time to test the method on every two clusters together to see whether we can distinguish the two clusters when they are put together in the graph of their metadata hierarchy.
In fact, when we look at the graph of metadata hierarchy of the two first clusters together, we notice, first of all, a loss of readability (*c.f.* Figure 5); but for those who know already each cluster, it's easy to identify the two thematic poles (or keywords) in the graph (*'projet'* and *'conseil'*) that are the two most common keywords between the two clusters; Thus, by deleting these two words from the graph we are able to distinguish directly our two different clusters.

When we came to the other two clusters and built their common hierarchy, we found a great ambiguity and it was very difficult to capture the two cluster out of the graph. This is because the two clusters have a lot of keywords in common; like (*'constitution'*, *'monde'*, *'pays'*, *'droit'*, etc).

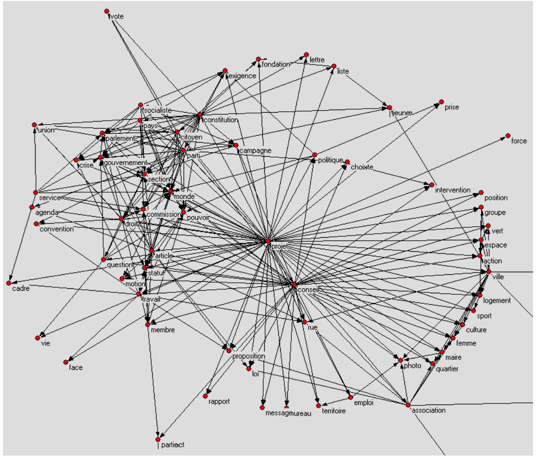

Figure 5: Part of the metadata hierarchy of the first two clusters together

5.1 The *semantic-hypertextual* graph

An essential and original goal of this paper is to verify the relationship between the topological clustering and the metadata hierarchy of web sites. Can we say that the clustering of web sites by means of their topological structure (i.e. inter-hypertext links) is also a semantic clustering?

In fact, to be able to answer this question, we will construct the *semantic-hypertextual* graph of web sites. This graph contains two types of nodes; the *web sites* nodes gathered into topological clusters and the main *keywords* nodes taken from the metadata hierarchy, and consequently, it also contains two types of links; the *topological* links between the web sites along with the *semantic* links that relate each web site with its keywords (We see in Figure 6 the web sites as small boxes, and the keywords as ovals). We used to see and study topological cartographies of web sites which represent the relations between the sites by means of their (outgoing/incoming) hypertext links; but this new graph is different. It gives us the ability to see the web sites and know the theme they're talking about since each keyword will be placed in the middle of the region of its main sites. Therefore, if we build the *semantic-hypertextual* graph of multiple clusters of web sites, we will be able to decide and verify the matching between the topological clustering and the semantic one.

6 Conclusion

We have seen that using the subsumption method in building a metadata hierarchy for a

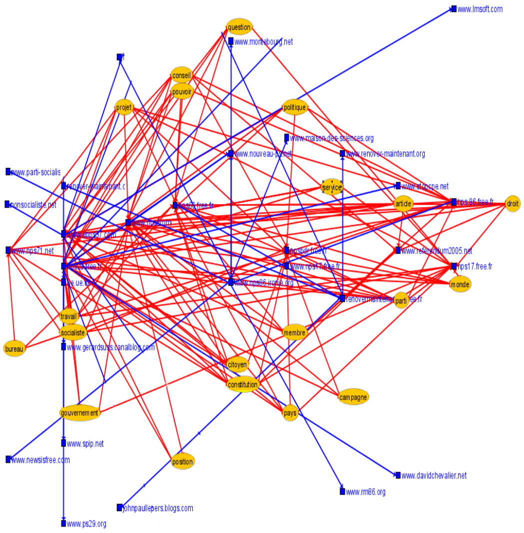

Figure 6: The *semantic-hypertextual* graph of the first cluster

set of web sites gives good results. We showed how we can also benefit from this hierarchy to compare multiple clusters by analyzing the graph of their shared content-oriented hierarchy and trying to find each cluster in it. The results prove that when the clusters are very different from each other, we will be able to find their keywords clearly separated in the shared graph, but the more these clusters have keywords in common, the much more difficult is to distinguish them in the shared graph. The method of grouping the synonymous words is still on the way to be applied, and we think that it will give unexpected results (some strange and some interesting). Designing the *semantic-hypertextual* graph of multiple clusters was an original and a new approach. It helped us determine if the topological clustering of web sites that depends on calculations made on the hypertext links (e.g. the *Hubs & Authorities*) is also correct from the semantic point of view.

7 Acknowledgments

We would like to thank all the members of the RTGI project for their great contribution in this paper.

References

[1] M. Sanderson, and B. Croft, "Deriving concept hierarchy from text", *In Proceedings of the ACM Sigir'99*, 1999, pp. 206-213.

[2] D. Lawrie, and B. Croft, "Discovering and Comparing Topic Hierarchies", *In Proceedings of RIAO*, 2000.

[3] P. Cimiano, A. Hotho, and S. Staab, "Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis", *Journal of Artificial Intelligence Research, Volume 24*, 2005, pp. 305-339.

[4] E. Stoica, and M. A. Hearst, "Nearly-automated metadata hierarchy creation", *In HLT-NAACL*, Short papers, 2004, pp. 117-120.

[5] S. Rydin, "Building a hyponymy lexicon with hierarchical structure", *ACL'02, SIGLEX Workshop on Unsupervised Lexical Acquisition*, University Pennsylvania, USA, 2002.

[6] S. A. Caraballo, "Automatic construction of a hypernym-labeled noun hierarchy from text", *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999, pp. 120-126.

[7] T. Hamon, A. Nazarenko, and C. Gros, "A step towards the detection of semantic variants of terms in technical documents", *In Proceedings of the Eighteenth International conference on Computational Linguistics (Coling'98)*, Montréal, 1998, pp. 498-504.

[8] F. Ghitalla, E. Diemert, C. Maussang, and F. Pfaender, "TARENTe: an Experimental Tool for Extracting and Exploring Web Aggregates", *IEEE International Conference on information & Communication Technologies: From Theory to Applications*, Damascus, Syria, 2004.

[9] D. Lin, "Using syntactic dependency as local context to resolve word sense ambiguity", *In Proceedings of ACL/EACL-97*, Madrid, Spain, 1997, pp. 64-71.

[10] G. Nenadic, I. Spasic, and S. Ananiadou, "Automatic Discovery of Term Similarities Using Pattern Mining", *In Proceedings of CompuTerm 2002*, Taipei, Taiwan, 2002, pp. 43-49.