

Learning Topic Hierarchies and Thematic Annotations from Document Collections

Hermine Njike Fotzo, Patrick Gallinari

LIP6

8, rue du capitaine Scott 75015 Paris France
{Hermine.Njike-Fotzo, Patrick.Gallinari}@Lip6.fr

1. Introduction

Large textual and multimedia databases are now widely available but their exploitation is restricted by the lack of meta-information about their structure and semantics. Many such collections like those gathered by most search engines are loosely structured. Some have been manually structured, at the expense of an important effort. This is the case of hierarchies like those of internet portals (Yahoo, Open Directory, LookSmart, etc) or of large collections like MEDLINE: documents are gathered into topics, which are themselves organized into a hierarchy going from the most general to the most specific [7]. Hypertext multimedia products are another example of structured collections: documents are usually grouped into different topics and subtopics with links between the different entities. Generally speaking, structuring collections makes easier navigating the collection, accessing information parts, maintaining and enriching the collection. Manual structuring relies on a large amount of qualified human resources and can be performed only in the context of large collaborative projects like e.g. in medical classification systems or for specific commercial products. In order to help this process it would be needful to rely on automatic or semi-automatic tools for structuring document collections.

In this context, we study here how to automatically structure collections by deriving concept hierarchies from a document collection and how to automatically generate from that a document hierarchy. The concept hierarchy relies on the discovering of “specialization/generalization” relations between the concepts which appear in the documents of a corpus. Concepts are automatically identified from the set of documents. The proposed method can also create “specialization/generalization” links between documents and document parts. It is a technique for the automatic creation of specific typed links between information parts. Such typed links have been advocated by different authors as a mean for structuring and navigating collections. It also associates to each document a set of themes representative of the main subjects treated in the document. The method is fully automatic and the hierarchies are directly extracted from the corpus, and could be used for any document collection. It could also serve as a basis for a manual organization.

The paper is organized as follows. In section 2 we introduce previous related work. In section 3, we describe our algorithm for the automatic generation of typed “specialization/generalization” relations between concepts and documents and the corresponding hierarchies. In section 4 we propose numerical criteria for measuring the relevance of our method. Section 5, describes experiments performed on a part of Looksmart and New Scientists hierarchies.

2. Related Work

The generation of hierarchies is a classical problem in information retrieval. In most cases the hierarchies are manually built and only the classification of documents into the hierarchy is automatic. Clustering techniques have been used to create hierarchies automatically like in the Scatter/Gather algorithm [3]. All these methods cluster documents according to their similarity. They cannot be used to produce topic hierarchies or to infer generalization/specializations relations. Recently, it has been proposed [12] to develop topic hierarchies similar to those found in e.g. Yahoo. As in Yahoo, each topic is identified by a single term. These term hierarchies are built from “specialization/generalization” relations between the terms, automatically discovered from the corpus. Croft, Lawrie, Sanderson [9, 12] propose to build term hierarchies based on the notion of subsumption between terms. Using related ideas, Krishna and Krishnapuram [8], propose a framework for modelling asymmetric relations between data. Vinokourov and Girolami [14] also propose a probabilistic model with a hierarchical structure for unsupervised organization of a collection into a hierarchy.

All these recent works associate the notion of concept to a term and rely on the construction of term hierarchies and the classification of documents within these hierarchies. Compared to that, we propose two original contributions. The first is the extension of these approaches to the construction of real concept hierarchy where concepts are identified by set of keywords and not only by a single term, all concepts being discovered from the corpus. These concepts better reflect the different themes and ideas which appear in documents, they allow for a richer description than single terms. The second contribution is the automatic construction of a hierarchical organization of documents also based on the “specialization/generalization” relation. This is described in section 3.

In section 4, we present new measures for similarity between concepts hierarchies. These measures could be compared to those proposed for comparing and matching ontologies although differences do exist in nodes representation [10].

For identifying concepts, we perform document segmentation into homogeneous themes. We used the segmentation technique of Salton [11] which relies on a similarity measure between successive passages in order to identify coherent segments. In [11], the segmentation method proceeds by decomposing texts into segments and themes. A segment is a bloc of text about one subject and a theme is a set of such segments. In this approach, the segmentation begins at the paragraph level. Then paragraphs are compared each other via a similarity measure.

3. The Models

3.1. Basic Ideas

This work started by studying the automatic derivation of typed links “specialization/generalization” between documents of a corpus. A link from document $D1$ to document $D2$ is of the type *specialization* (generalization from $D2$ to $D1$), if $D2$ is about specifics themes of $D1$. For example, $D1$ is about war in general and $D2$ is about the First World War in particular. This type of relation allows

building hierarchical organizations of the concepts present in the corpus which in turn allows for the construction of a hierarchical corpus organization.

In hierarchies like Yahoo!, the concepts used to organize documents are reduced to words. This gives only basic indications on the content of a document and the corresponding hierarchies are relatively poor. For this reason, we have tried to automatically construct hierarchies where each concept will be identified by a set of words. In order to do this, we need the knowledge of all themes present in the collection and of the specialization/generalization relations that do exist among them. From now, we will identify a concept to a set of keywords.

Our method is built around three main steps:

- Find the set of concepts of a given corpus
- Build a hierarchy (of type specialization/generalization) of these concepts
- Project the documents in the concepts hierarchy and infer typed links “specialization/generalization” between documents.

3.2. Algorithm

Concepts extraction from the corpus:

Our goal is to detect the set of concepts within the corpus and the words that represent them. For that, we extend Salton work on text segmentation: first we decompose a document into semantic themes as in Salton’s method [11]. Each document is then decomposed in a set of semantic themes and then all the themes are clustered to retain the minimal set of themes that ensure a correct coverage of the corpus. We find for each concept the set of words that represent the concept. A concept is represented here by its most frequent words.

Building the concept hierarchy:

In this step we try to detect the “specialization/generalization” relations between extracted concepts in order to infer the concept hierarchy. We describe two methods for that:

- Method 1 : the first method detects these relations between concepts by exploiting a term hierarchy using Croft and Sanderson subsumption method (term $t1$ subsumes term $t2$ if the following relation holds : $P(t1/t2) > t$ and $P(t2/t1) < P(t1/t2)$ where t is a preset threshold.) [12]. Then we create a concept hierarchy as follows: For each couple of concepts, we compute from the terms hierarchy the percentage x of words of concept $C2$ generalized by words of concept $C1$ and y the percentage of words of $C1$ generalized by words of $C2$. If $x > S1 > S2 > y$ ($S1$ and $S2$ are thresholds) then we deduce a relation of specialization/generalization between these concepts ($C1$ generalizes $C2$).
- The second method consists in estimating $P(Ci|Cj)$ for all pairs of concepts and apply Croft subsumption definition directly to the extracted concepts. We have tested two ways for estimating these conditional probabilities:
 - Method 2 : After the segmentation process (clustering of paragraphs), we know the documents the paragraphs of a concept com from. We can then attach these documents to the concept nodes. Then we can estimate $P(Ci|Cj)$ by counting: $P(Ci|Cj) = \text{number of documents common to nodes } Ci \text{ and } Cj / \text{number of documents in node } Cj$. We apply transitivity to end the hierarchy construction.
 - Method 3 : Another way to attach documents to concept is by estimating $P(C|d)$, $P(C|t)$. We do this via an simple EM algorithm. We can then estimate $P(Ci|Cj)$ by counting like above.

After that, we have a hierarchical organization of concepts and the assignment of indexed documents to the nodes in the hierarchy. One document may belong to different nodes if it is concerned with different concepts.

4. Evaluations Measures

Evaluating the relevance of a concept or document hierarchy is a challenging and open problem. Evaluations on user groups generally give ambiguous and partial results while automatic measures only provide some hints on the intrinsic value of the hierarchies. However, for avoiding at this stage the heavy process of human evaluation, we resort to automatic criteria to judge the quality of learned hierarchies. We therefore propose two measures of similarity between hierarchies. This will allow to compare the coherence of our automatic hierarchies to reference manual hierarchies (here a part of LookSmart and NewScientist hierarchies), but will not provide an indication of its absolute quality, neither will it tell us which hierarchy is the best. We then propose another measure, which captures how a hierarchy reflects the specialization/generalization property.

4.1. Similarity Measures

4.1.1. A measure based on the inclusion of a hierarchy into another hierarchy: documents in the hierarchy are said to share a relation of “Brotherhood” if they belong to the same node or a “Parent-child” relation if they belong to nodes of the same branch. The first measure of similarity we propose is based on the mutual inclusion degree of hierarchies. The inclusion degree of hierarchy A with respect to hierarchy B is: $Inclusion(A,B) = (N_f + N_p) / (|F_A| + |P_A|)$, Where N_f is the number of couples of “brothers” in A which belong to B . N_p is the number of couples “parents-child” in A which belong to B . $|F_A|$ is the number of couples of “brothers” documents in A . $|P_A|$ is the number of couples of “parents-child” in A . Finally, the similarity between A and B is the average of their mutual inclusion: $Similarity(A, B) = (inclusion(A, B) + inclusion(B, A)) / 2$

4.1.2. A measure based on Mutual Information between Hierarchies: In [4], the authors propose a similarity measure to compare two clustering algorithms. Let X and Y be the labelling (class) of all elements from a dataset according to two different clustering algorithms, X_i be the label for the i^{th} cluster in X , $P_X(C = k)$, that an object belongs to cluster k in X , and $P_{XY}(C_X = k_x, C_Y = k_y)$ the joint probability that an object belongs to cluster k_x in X and to cluster k_y in Y .

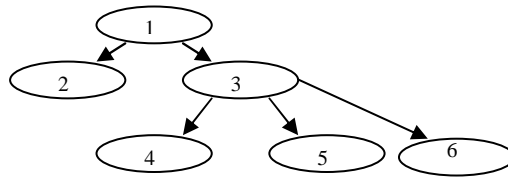
To measure the similarity of the two clustering methods, the authors propose to use the mutual information between the two probability distributions: $MI(X,Y) = \sum_{i \in C_X} \sum_{j \in C_Y} P_{XY}(C_X = i, C_Y = j) * \log [(P_{XY}(C_X = i, C_Y = j)) / (P_X(C_X = i) * P_Y(C_Y = j))]$. If MI is normalized between 0 and 1 the more $MI(X, Y)$ is close to 1 the more similar are the two set of clusters and therefore the methods.

In the case of hierarchical clustering, for measuring the similarity between two hierarchies, we need to measure how objects are grouped together (inside the hierarchy nodes) and to measure the similarity of the relations “parent-child” between objects in the two hierarchies. For simplifying the description, we will first consider that in each clustering one object may belong only to one cluster. The extension to the case where one object may appear in different nodes is obvious.

Using notations similar to [4], for clustering X , X_i will be here a branch of the hierarchy which contains the objects of its terminal node, and all “parents-child” couples (a,b) on the branch, with “ b ” an object of the terminal node on X_i and “ a ” an object from any other node on X_i . Given n objects in the nodes of a hierarchy, we have here to compare the distribution in the two hierarchies of $n + n * (n-1)$ objects (the initial objects and the potential couples). However, many couples do not exist in the hierarchy. Let us then create a fictive node which contains all the couples of documents that do not share a “parents-child” relation. From then on, we are in the same situation as in [4] if we consider the distribution on objects and couples in the hierarchy instead of the distribution on objects only in [4]. We can apply the measure proposed in [4] to compare hierarchies. In our experiments, we consider both the clustering on documents with $MI_D(X_D, Y_D)$ the mutual information between two hierarchies measured only on documents and the clustering on documents couples with $MI_{P.C}(X_{P.C}, Y_{P.C})$ the mutual information measured on all couple Parent-Child of the hierarchies. The mutual information between hierarchies X and Y is then, $MI(X, Y) = \alpha * MI_D(X_D, Y_D) + (1 - \alpha) * MI_{P.C}(X_{P.C}, Y_{P.C})$, here α is a parameter which allows giving more importance to the grouping of documents in similar nodes or to the hierarchical P-C relations between documents.

4.2. Quantification of “specialization/generalization” quality of a hierarchy

In [9], the authors propose to measure the quality of a hierarchy the mutual information between root terms and the rest of the vocabulary: $I(T, V) = \sum_{t \in T} \sum_{v \in V} P(t, v) * \log (P(t, v) / (P(t) * P(v)))$, where T are roots terms and V the other terms of the vocabulary.



For the above hierarchy $I(T, V)$ gives an idea on how the term 1 predicts the rest of the vocabulary {2,3,4,5,6}, but not on how term 3 is generalizes its children. We propose a new measure to quantify the specialization/generalization property of a hierarchy, which takes into account the generalization property of each node.

Conditional entropy measures the uncertainty on a variable knowing another: $H(Y|X) = -\sum_x \sum_y P(x, y) * \log(P(y|x))$. In the subsomption framework if term x generalizes term y , the uncertainty on x knowing y is low. We chose the conditional entropy to quantify the generalization property of a term with respect to its children. We define the generalization ability of term t by :

- $TermIndexGeneralization(t, \{f\}) = \sum_f - P(t, f) * \log(P(t|f))$, where $\{f\}$ denotes t children. The lower is the index the more t will be a good generalization of its children $\{f\}$
- The generalization index of a hierarchy is : $HierarchyIndexGeneralization = \sum_t TermIndexGeneralization(t, \{f\})_t$.

The measure presented above is designed for term hierarchies. It can be easily extended to concept hierarchies.

5. Experiments and Results

5.1. Data

The data we used for our experiments are a part of the www.looksmart.com and www.newscientist.com sites hierarchies. First, we extracted a sub-hierarchy of LookSmart consisting of about 100 documents and 7000 terms about artificial intelligence. In a second experiment, we extract a sub-hierarchy of New-Scientist site consisting of about 700 documents. New-Scientist Web site is a weekly science and technology news magazine which contains all the latest science and technology news. Here the sub-hierarchy is heterogeneous sub-hierarchy whereas LookSmart data concern only AI. Documents are about AI, Bioterrorism, cloning, Dinosaurs, and Iraq. For each theme there are sub-categories concerning specifics aspects of the theme. In both cases, we compare the document hierarchies induced by our method and the term hierarchies to the original hierarchies, using the methods described in section 3.

5.2. Experiments and Results

5.2.1. Example of extracted concepts

Compared to the Looksmart hierarchy with five categories, the hierarchy derived by our algorithm on the same corpus is larger and deeper. Categories are more specific and the algorithm discovers many more thematics. For example, many sub-categories emerge from the “Knowledge Representation” area like: ontologies, building ontologies, KDD (where papers deal with data representation for KDD, etc.. In the same way, “Philosophy-Morality” is subdivided in many categories like AI definition, Method and stakes, risks and so on. Table 2 shows some examples of extracted themes

| LookSmart | |
|-----------|---|
| 1 | definition AI intelligence learn knowledge solve build models brain Turing Test thinking machine |
| 2 | informal formal ontology catalog types statements natural language names axiom definition logic |
| 3 | FCA techniques pattern relational database data mining ontology lattice categorie |
| 4 | ontology Knowledge Representation John Sowa categories artificial intelligence philosophers Charles Sanders Peirce Alfred North Whitehead pioneers symbolic logic |
| 5 | system KR ontology hierarchy categories framework distinction lattice chart |

The table shows examples of five concepts extracted from the Looksmart corpus. Each concept is identified by a set of representative keywords. The algorithm discovers a generalization/ specialization relation between concepts (2, 3), (2, 4), (2, 5)

5.2.2. Similarity between Hierarchies

For more details about the method definition (see section 3):

- Croft hierarchy is the hierarchy term obtain by Croft method based on subsumption
- Method 1 is concepts hierarchy build by projection of concepts on Croft hierarchy
- Method 2 is the direct application of subsumption definition to concepts, with documents affectation derived by segmentation results
- Method 3 the direct application of subsumption definition to concepts, with documents affectation derived to the estimation of $P(\text{Concept} | d)$ by EM algorithm. For the two last methods $P(\text{concept1}|\text{concept2})$ is by counting.

| | Croft Hierarchy | Method 1 | Method 2 | Method 3 |
|---------------------|-----------------|----------|----------|----------|
| LookSmart | | | | |
| Inclusion | 0.4 | 0.46 | 0.65 | 0.7 |
| Mutual Information | 0.3 | 0.6 | 0.7 | 0.7 |
| NewScientist | | | | |
| Inclusion | 0.3 | 0.2 | 0.6 | 0.6 |
| Mutual Information | 0.2 | 0.2 | 0.65 | 0.67 |

If we compare the document hierarchies built from term hierarchies with that of LookSmart, we see (table, column Croft hierarchy) that inclusion similarity is **0.4** and the mutual information is **0.3**. Both hierarchies use terms to index and organize documents. However, the term hierarchy uses all terms in the collection, whereas LookSmart uses a much smaller vocabulary. Therefore the hierarchy term is very large compared to LookSmart. Nevertheless some groups of documents are still common to the two hierarchies.

The hierarchy obtained with our method by organizing documents according to automatically discovered concept hierarchies is large compared to the originals ones. The greater width of our hierarchy is due to the fact that some themes detected through corpus segmentation are not present in original hierarchies which exploit simpler conceptual representations. Nevertheless, the similarity between our hierarchy and LookSmart's is quite high for the three methods, with better result for method3. But the similarity between our hierarchy and New-Scientist one is lower. This result points out the weakness of the subsumption method between document terms, when the data are heterogeneous. Remember method1 rely on terms hierarchy and it's not the case for method 2 and 3. Computing directly subsumption between concepts gives a hierarchy much more similar than the original one. One way to reduce further the influence of vocabulary heterogeneity would be to consider synonyms in the computation of $P(\text{term1}|\text{term2})$.

Note that the similarities obtained by organizing document around automatically discovered concept hierarchies are much higher than those obtained with the term hierarchies. These experiments shed some light on the algorithm behaviour. The hierarchies we obtain are coherent (particularly those obtained with our two last methods) with LookSmart and New-Scientists hierarchies.

5.2.3. Specialization/generalization property of hierarchies

For this measure (section 4.2) of the generalization capacity, the lower the index value is, the better the method is.

| | LookSmart | Croft Term Hierarchy | Method 1 | Method 2 | Method 3 |
|---|--------------|----------------------|----------|----------|----------|
| Specialization / generalization measure | 41.53 | 20.62 | 15.2 | 3.71 | 3.8 |
| | NewScientist | Croft Term Hierarchy | Method 1 | Method 2 | Method 3 |
| Specialization / generalization measure | 50.12 | 45.2 | 32.11 | 11.57 | 10.87 |

Table 3

Results in table 3 shows that the original hierarchies have a low generalization capacity. The organization produced by method 2 or 3 seems to be the best with respect to the expected generalization property. Nevertheless these nice results should still considered carefully since what matters is for example the way a real user understands the different hierarchies. To complete these analyses we shall extend these experiments by additional evaluations.

6. Conclusions

We have described a method to automatically generate a hierarchical structure from a document collection. The same method can be used to build specialization/generalization links between documents or document parts and to annotate documents with the hierarchically organized concepts. We have also introduced three new numerical measures for the open problem of comparing and evaluating such hierarchies. Two of them give an indication on the proximity of two hierarchies and allow measuring the coherence of two different hierarchies. On the other hand, they do not say anything on the intrinsic quality of the hierarchies. The third measure quantifies how much a hierarchy respects the "specialization/generalization" property.

- Our method applied to LookSmart and New-Scientists data gives interesting results. The experiments also show that our concepts hierarchies are nearer to original hierarchies than a reference method which automatically builds term hierarchies. Further experiments on different collections and on a larger scale are needed to confirm this fact.

7. Bibliography

- [1] J. Allan. Automatic hypertext link typing. Proceeding of the ACM Hypertext. March 1996 Conference, Washington, DC pp.42-52.
- [2] C. Cleary, R. Bareiss. Practical methods for automatically generating typed links. Hypertext '96, Washington DC USA
- [3] D. R. Cutting, D. R. Karger, J. O. Pedersen, J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In ACM SIGIR, 1992.
- [4] T. Draier, P. Gallinari. Characterizing Sequences of User Actions for Access Logs Analysis, User Modeling 2001, LNAI 2109.
- [5] B. Gelfand, M. Wulfekuhler, W. Punch. Discovering Concepts in Raw Texts: Building Semantic Relationship Graphs. In ICML/AAAI workshop on learning for text categorization, 1998.
- [6] M. Hearst. TextTitling : Segmenting Text into multi-paragraph Subtopic Passages. Computational Linguistics, pp. 33-64, 1997.
- [7] G. Källgren. Automatic Abstracting on Content in text. Nordic Journal of Linguistics. pp. 89-110, vol. 11, 1988.
- [8] K. Krishna, R. Krishnapuram. A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management. Atlanta, Georgia, USA. Pp.571-573
- [9] D. Lawrie B. Croft, A. Rosenberg Finding Topic Words for Hierarchical Summarization
- [10] A. Maedche, S. Staab. Measuring Similarity between Ontologies. European Conference on Knowledge Acquisition and Management, EKAW-2002, Madrid, Spain, LNCS/LNAI2473, Springer 2002, pp 251-263.
- [11] G. Salton, A. Singhal, C. Buckley, M. Mitra: Automatic Text Decomposition Using Text Segments and Text Themes. Hypertext 1996: 53-65
- [12] Mark Sanderson, Bruce Croft. Deriving concept hierarchies from text. In Proceedings ACM SIGIR Conference '99, 206-213, 1999.
- [13] Randall Trigg. A network-based approach to text handling for the online scientific community. University of Maryland, Department of Computer Science, Ph.D dissertation, November 1983.
- [14] A. Vinokourov, M. Girolami. A Probabilistic Hierarchical Clustering Method for Organizing Collections of Text Documents. Proceedings of the 15th International Conference on Pattern Recognition (ICPR'2000), Barcelona, Spain. IEEE computer press, vol.2 pp.182-185