

Ontology from Hierarchical Structure in Text

F. Murtagh, Department of Computer Science, Royal Holloway,
University of London Egham TW20 0EX, UK (fionn@cs.rhul.ac.uk)

S. McKie, Department of Media Arts, Royal Holloway, University of London
Egham TW20 0EX, UK (sm@tripos.biz)

J. Mothe, Institut de Recherche en Informatique de Toulouse
118, Route de Narbonne, 31062 Toulouse Cedex 04, France
mothe@irit.fr

and

K. Englmeier, Fachhochschule Schmalkalden, Blechhammer
98574 Schmalkalden, Germany (k.englmeier@fh-sm.de)

August 12, 2007

Abstract

We study the notion of hierarchy in the context of visualizing textual data and navigating text collections. A formal framework for “hierarchy” is given by an ultrametric topology. This provides us with a theoretical foundation for concept hierarchy creation. Frequency of occurrence (terms, texts) data is embedded in a metric space by correspondence analysis, from which ultrametric properties of the data are explored and exploited. A major objective is *scalable* annotation or labeling of concept maps, for summarizing information and supporting navigation. We exemplify our approach using (i) a single text subdivided into successive parts (for which we provide an interactive demonstrator), and (ii) a collection of texts, among them television series scripts, providing an important template for varied semi-structured text stores. We provide comparisons with other visualization approaches.

Categories and Subject Descriptors: H.5 (Information interfaces and presentation), I.5.3 (Clustering), H.5.2 (User interfaces), I.7.2 (Document preparation), H.3 (Information storage and retrieval)

1 Introduction

1.1 Organization of the Article

In section 1, we review the application domain of visual user interfaces and their role in information space navigation. We discuss how structuring of summarized

information is needed. We review related work. We then discuss the semi-structured data that is a useful exemplar of various application domains.

In section 2 we discuss background technologies for later work presented here, illustrating various aspects of hierarchical structure in textual data.

Section 3 covers how we derive a dominance relationship between terms.

In section 4 we present a range of examples, and comparisons with two tag cloud tools.

1.2 Visual, Interactive User Interfaces for Supporting Information Space Navigation

Since the mid-1990s we built visual interactive maps of bibliographic and database information at Strasbourg Astronomical Observatory, and some of these, with references, are available at Murtagh [2006c]. The semi-structured display used by us was based on the Kohonen self-organizing feature map. A comprehensive view of the latter can be found at websom.hut.fi [Kohonen et al. 2000].

A more recent development has been tag clouds. McKie [2007] discusses examples and provides an online system for creating of filmscript term “clouds”. He discusses similar tools (e.g., TagCrowd, www.tagcrowd.com; Zoomcloud, zoomclouds.egrupos.net). Similar tag clouds are commonly used to present information in large data repositories (e.g. flickr, www.flickr.org).

The motivation for such tools is to have (possibly interactive) annotated maps to support information navigation. Prominent terms are graphically presented and can be used to carry out a local search. In some cases, the location

of terms is important, in particular in the case of the Kohonen map. The automated annotation of such information maps is not easy. Often the basis for display font size and sometimes even for location on the maps is simply frequency of term occurrence. The work presented in this article aims at taking more available information into account, leveraging interrelationships in textual content and thereby semantic content.

Visualization is often an important way to elucidate semantic heterogeneity for the user. Visual user interfaces for ontological elucidation are discussed in Murtagh et al. [2003], with examples that include interactive, responsive information maps based on the Kohonen self-organizing feature map; and semantic network graphs. A study is presented in Murtagh et al. [2003] of client-side visualization of concept hierarchies relating to an economics information space. The use of “semantic road maps” to support information retrieval goes back to Doyle [1961]. Motivation, following Murtagh et al. [2003], includes the following: (i) Visualization of the semantic structure and content of the data store allows the user to have some idea before submitting a query as to what type of outcome is possible. Hence visualization is used to summarize the contents of the database or data collection (i.e., information space). (ii) The user’s information requirements are often fuzzily and ambiguously defined at the outset of the information search. Hence visualization is used to help the user in his/her information navigation, by signaling the relationships between concepts. (iii) Ontology visualization therefore helps the user before the user interacts with the information space, and during this interaction. It is a natural enough pro-

gression that the visualization becomes the user interface.

1.3 Ontology

“Ontologies are often equated with taxonomic hierarchies of classes ... but ontologies need not be limited to such a form” [Gruber 2001]. Gruber is cited in Gómez-Pérez et al. [2004] as characterizing an ontology as “an explicit specification of a conceptualization”. In Wache et al. [2001], ontologies are motivated by semantic heterogeneity of distributed data stores. This is also termed data heterogeneity and is counterposed to structural or schematic heterogeneity. Ontologies are motivated by Wache et al. [2001] “for the explication of implicit and hidden knowledge”, as “a possible approach to overcome the problem of semantic heterogeneity”. So, ontologies may help with integration of diverse, but related, data; or they may help with clarifying or disambiguating distinctions in the heterogeneous data. Ontologies are likely to be of immediate help in supporting querying. For example, the query model may be based on the ontology (or ontologies) used.

There is extensive activity on standards and software, relating more to the above-mentioned schematic rather than semantic heterogeneity, and a useful survey of this area is Denny [2004]. Denny takes an ontology in a broad-ranging view as a knowledge-representation scheme.

1.4 Tools: Linguistic Analysis, Ancillary Databases, Algebraic Structures

Linguistic analysis tools, and ancillary databases, have often been used in ontology construction, and we will now briefly review these options.

Much work on text mining avails of statistical analysis (mostly in the form of frequency of occurrence and cross-tabulation data) and, to a greater or lesser degree of linguistic analysis. The latter may include relation finding in order to target verbs and their associated grammatical arguments or terms [Schutz and Buitelaar 2005]. Linguistic analysis is used for “augmenting” [Gillam et al. 2005] or providing “semantic enrichment” [Xiaomeng Su and Gulla 2006], or to be “trimmed and enriched” [Navigli and Velardi 2004]. Linguistic analysis may go as far as the use of already existing ontologies or semantic nets (implying “semantic enrichment”: [Xiaomeng Su and Gulla 2006]) such as WordNet (e.g. [Navigli and Velardi 2004]); or use more direct syntactic information such as word inclusion in multiple word expressions [Navigli and Velardi 2004]. (Pioneering work on term completion was due to Spärck Jones [1971].) The syntactic approach, – generalization and equivalence relations (approximated by “is-a” and clustering, respectively) – is approached by De la Higuera and Daniel-Vatonne [1996] in terms of programming language semantics.

In section 2.1 below we introduce a new way of quantifying inherent hierarchical structure as one type of semantic information. Apart from information measures, Ahmad et al. [1996] defines a document “weirdness index” as a simple coefficient of specialist versus general terminology. A different approach to co-

hension analysis has been on the basis of path lengths between terms in semantic networks such as WordNet, or more generally on the graph of lexical, referential, or verb links (see [Chan 2004; 2006]). A semantic network also underpins the work of Tucker and Spärck Jones [2005]. When the graph (i.e., network) is restricted to dominance relationships, then it is possible that we have a lattice. A lattice is a partially ordered set (poset) such that every pair of elements has a least upper bound (see [Davey and Priestley 2002]). The application of lattices to the semantics of, for example, text allows for Formal Concept Analysis (FCA), [Cimiano et al. 2005].

Term extraction and use of parsers are commonly used for ontology learning: “The state-of-the-art is mostly to run a part-of-speech tagger over the domain corpus used for the ontology learning task and then to identify possible terms by manually constructing ad-hoc patterns ... whereas more advanced approaches to term extraction for ontology learning build on deeper linguistic analysis ...” [Buitelaar et al. 2005].

Ahmad and Gillam [2005] develop a semi-automated approach using text with no markup. Multiword expressions are determined, and frequency of occurrence information is used to point to term or phrase importance. A stop list is used to avoid irrelevant words. Part of speech analysis is not used. A semantic net is formed to allow development of the ontology elements. Fuzzy inference is used by Lee et al. [2007].

Abou Assali and Zanghi [2006] use syntactic part of speech tagging to determine the nouns. These authors retain sufficiently frequent nouns. They apply

the notion of weak subsumption: if – for the most part – a word is in a text that another is in, and not vice versa, then this leads to a hierarchical relationship.

Chuang and Chien [2005] assert that multiway trees are appropriate for concept hierarchies, whereas binary trees are built using hierarchical clustering algorithms. Hence they modify the latter to provide more appropriate output. (A formal approach for mapping a binary hierarchical classification tree onto a multiway hierarchy is described in Murtagh [2007].)

A hierarchical clustering has often been used to represent an ontology. Note that this is usually not a concept hierarchy. A concept hierarchy is based on a subsumption relationship between terms, whereas a hierarchical clustering is an embedded set of clusters of the term set. Later in this article (section 3), we show a way to derive a concept hierarchy, involving subsumption of terms, from a hierarchic clustering.

A hierarchic clustering is typically a binary, rooted, terminal labeled, node ranked tree, and a concept hierarchy is typically a multiway, rooted, terminal and non-terminal labeled, ranked tree. By starting with the former (binary) tree representation, we have an extensive theoretical and formal arsenal at our disposal, to represent the main lines of what we need to do, and to help to avoid being overly reliant on user parameter-based aggregation of subsystem components and tools. As seen later in this work, we start by laying the foundations of our perspective by basing this on binary trees, and later proceed to the multiway tree. An alternative approach can be found in Ganesan et al. [2003], where similarities or distances on trees are redefined and re-axiomatized for the case

of multiway trees.

An alternative representation for an ontology is a lattice and Formal Concept Analysis, already noted, is a methodology for the analysis of such lattices. If we have a set of documents or texts, I , characterized by an index term set J , then as Janowitz [2005] shows, hierarchical clustering and FCA are loosely related. Hierarchical clustering is based on pairwise distances or dissimilarities, $d : I \times I \rightarrow \mathbb{R}^+$ (\mathbb{R}^+ is the set of non-negative reals). FCA is based on partially ordered sets (posets) such that there is a dissimilarity $d : I \times I \rightarrow 2^J$ (2^J is the power set of the index terms, J). There are thus linkages and also differences between FCA and the hierarchical clustering approach that we use in this work: see Murtagh et al. [2007] for a discussion of how and where a hierarchy derived from an FCA approach differs from one derived from a stepwise hierarchical clustering algorithm.

Other approaches (rule-based; machine learning approaches, etc.; layered, engineering, approaches with maintenance management – see Maedche [2006]) are also available. One difficulty with the engineering of such approaches is that there is an ad hoc understanding of the problem area, and often there is dependence on somewhat arbitrary threshold and selection criteria that do not generalize well.

Our approach formalizes the problem area – the information space – in terms of its local or global topology. Where we do have selection criteria, such user interaction is at the application goal level.

1.5 Semi-Structured Text

Chafe [1979] considers linear versus hierarchical (e.g., at sentence, paragraph, section, etc. levels) organization of text, in the context of studying narrative in its role as expressing past experience. Chafe used a 7-minute 16 mm color movie, with sound but no language, and collected narrative reminiscences of it from human subjects. Chafe argues in favor of a “flow model”, i.e. a “flow of thought and the flow of language” which is not particularly interested in structure of any kind. In our work, based on film or television movie scripts, we have and avail of given scene boundaries. For Chafe, and others basing their work on a similar principle such as Hearst [1994], this was not the case and instead they based their work on the human thinking that lay behind the recorded narrative. It will be informative and very useful for us to avail instead of the structure that is provided by a film or television program script.

There are literally thousands of film scripts, including for television programs, for all genres, available and openly accessible on the web (e.g. [IMSDb](#), [Internet Movie Script Database](#)). A film script is composed of a succession of scenes, each of which has a header (often in upper case, and indented) indicating internal or external, day or night, location and other metadata, together with transition (“cut to:”, “sound cue”) and beginning and end details. The variable length scene itself contains dialog between characters, and/or action description. Supporting information retrieval of movie scripts is of importance in the writing and rewriting process. Indeed machine learning algorithms have been directly applied to scripts themselves to predict later commercial success [Glad-

well 2006]. Analysis, retrieval and use of scripts provide a paradigmatic case for medical report handling, and scenario analysis in organization and management. The scripts may provide a more malleable basis for reshaping and restructuring content in order to support interactive training and learning environments, as well as the full gamut of interactive media in entertainment.

For all these reasons the television program scripts, used later, represent (technically) ideal and (application-wise) important exemplars for us. As we will see below, we make full use of whatever structure is available to us in such data.

1.6 Innovation in Our Approach

Our approach is distinguished by the following.

1. It is unsupervised rather than a supervised or learning approach (cf. the title and theme of [Buitelaar et al. 2005]. We generally succeed in having our processing pipelines automated and not requiring user parameters.
2. Our approach is statistical in the sense of being based on frequency of occurrence or presence/absence characterization of the data; and it is based on the topology of the information space. This helps in achieving our goal of a comprehensive and complete processing pipeline, bypassing the need for an “ontology engineering architecture” as such (cf. [Navigli and Velardi 2004; Velardi et al. 2997]).
3. A Euclidean embedding is often used for filtering (so that the higher

eigenvalue-ranked axes account for the more important sources of variation, hence information, in one's data) and display. Examples include latent semantic indexing, (metric or non-metric) multidimensional scaling, etc. In our use of correspondence analysis in this article, we avail of a full dimensionality embedding. Hence we avoid any limitations related to less than full dimensionality approximation. Our motivation is (i) the data normalization that is part and parcel of the embedding; and (ii) the ease with which we use the Euclidean embedded data to induce a hierarchical clustering.

2 Three Metrics: Chi Squared, Euclidean and Ultrametric

2.1 Quantifying Hierarchical Structure

We will briefly review work which has led to the results described in this article.

A basic issue for us in the context of finding an ontology is whether or not any text object or repository document has any hierarchical structure to begin with. Alternatively we may wish to consider the issue of whether or not a document has sufficient inherent hierarchical structure to warrant further investigation. We could approach this problem by fitting a hierarchy, and there are many algorithms for doing so (such as any hierarchical clustering algorithm; de Soete [1986] describes a least squares optimal fitting approach).

In Murtagh [2004] we show how we can quantify the extent of hierarchical

structure in text. In keeping with the standard approach in multivariate data analysis, a hierarchy is defined (without loss of generality) as a binary, rooted, sometimes labeled, tree. Such a tree defines an ultrametric topology and, reciprocally, the tree is a representation for the ultrametric space. (Knapp [1988] uses the term “hierarchical” to mean self-similar.) In Murtagh [2006a, 2006b] further explorations are described in the assessment of inherent hierarchicity or ultrametricity in text. We will return to the characterization of ultrametric spaces in section 3.2 below.

2.2 Euclidean Embedding

In our use of free text, a mapping into a Euclidean space gives us the capability to define distance in a simple and versatile way. In correspondence analysis [Murtagh 2005], the texts we are using provide the rows, and the set of terms used comprise the column set. In the output, Euclidean factor coordinate space, each text is located as a weighted average of the set of terms; and each term is located as a weighted average of the set of texts. (This simultaneous display is sometimes termed a biplot.) So texts and terms are both mapped into the same, output coordinate space. This can be of use in understanding a text through its closest terms, or vice versa.

A commonly used methodology for studying a set of texts, or a set of parts of a text (which is what we will describe below), is to characterize each text with numbers of terms appearing in the text, for a set of terms. The χ^2 distance is an appropriate weighted Euclidean distance for use with such data [Benzécri

1979; Murtagh 2005]. Consider texts i and i' crossed by words j . Let k_{ij} be the number of occurrences of word j in text i . Then, omitting a constant, the χ^2 distance between texts i and i' is given by $\sum_j 1/k_j (k_{ij}/k_i - k_{i'j}/k_{i'})^2$. The weighting term is $1/k_j$. The weighted Euclidean distance is between the *profile* of text i , viz. k_{ij}/k_i for all j , and the analogous *profile* of text i' . (Our discussion is to within a constant because we actually work on *frequencies* defined from the numbers of occurrences.)

The relationship between the χ^2 distance and mutual information is explored in Benzécri [1979, Tome 1, pp. 216–218]. There is much formal similarity between the two: the χ^2 distance avails of a quadratic relationship, whereas mutual information uses the logarithm of a ratio. Benzécri discusses where one can be effectively substituted for the other. The singular value decomposition at the heart of correspondence analysis of course rests on the χ^2 distance.

Correspondence analysis allows us to project the space of documents (we could equally well explore the terms in the *same* projected space) into a Euclidean space. It maps the all-pairs χ^2 distance into the corresponding Euclidean distance.

For a term, we use the (full rank) projections on factors resulting from correspondence analysis. As noted, this factor space is endowed with the (unweighted) Euclidean distance.

2.3 Example: “Crime Scene Investigation” CSI 1-01 Pilot

As an exemplary data set, we use transcripts for the widely-aired CSI, “Crime Scene Investigation”, television series. The Pilot, 1-01, was originally aired on CBS on October 6, 2000. Eight seasons’ scripts are available for this, the original Las Vegas series [TWIZ, 2007]. In the Pilot, there are 50 scenes, with word counts ranging from 146 words to 676 words. In all there are 9934 words. There are 1679 unique words, greater than 1 letter in length, with lower case replacing upper case, and with punctuation ignored. We will use this 1679 unique word set.

The frequency of occurrence data crossing the 50 scenes and 1679 words is mapped, using correspondence analysis, into a space of inherent dimensionality 49: if n, m are respectively the numbers of rows or scenes, and columns or words, then the inherent dimensionality is $\min(n - 1, m - 1)$; the reason why 1 is subtracted from both is that the cloud of scenes and the cloud of words are both centered, giving a linear dependence. The origin is the average, expressing the hypothetical scene, or the hypothetical word, carrying no information.

In Figure 1, the scenes and words are located in the same embedding. The figure is interpreted in a natural, Euclidean, way, which is not the same as when we are presented with a frequency of occurrence data array. Defined on the basis of the frequency of occurrence array, we have the χ^2 distance between scenes and/or words. The output display in Figure 1 is a best planar view of a space endowed with the Euclidean metric. Both scenes and words have a “built-in” normalization. The one very important fact to keep in mind is that

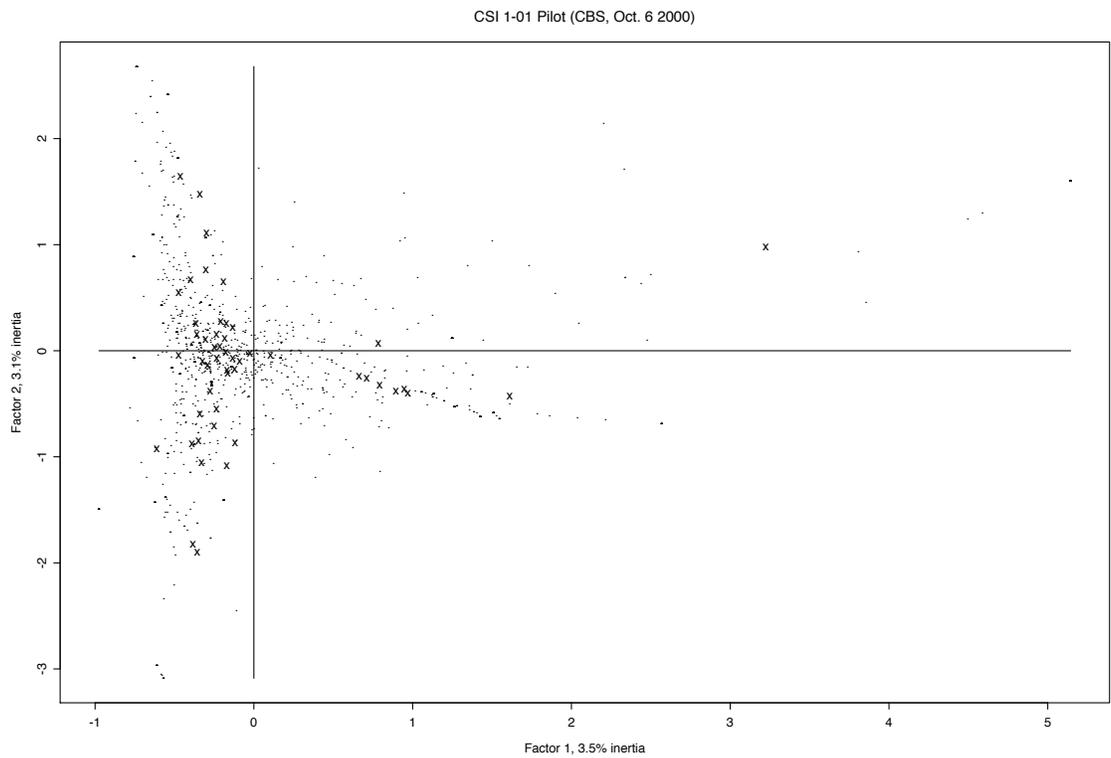


Figure 1: Correspondence analysis principal factor plane of projections of 50 scenes (each represented with an x), and 1679 characterizing words (each represented with a dot).

this is a best planar view of what is, in reality, a 49-dimensional space. The quality of the approximation involved in this is seen in the percentage inertia explained by these factors. Quite typically for correspondence analysis, the extent of approximation is weak. This is because less important factors or axes are “explained by”, or determined by, isolated, very particular, words (which thereby also determine the information content of particular scenes).

The relationship between scenes and words in Figure 1 is ultimately given by the dual space relationships [Murtagh 2005]:

$$F_\alpha(i) = \lambda_\alpha^{-\frac{1}{2}} \sum_{j \in J} f_j^i G_\alpha(j) \text{ for } \alpha = 1, 2, \dots, N; i \in I$$

$$G_\alpha(j) = \lambda_\alpha^{-\frac{1}{2}} \sum_{i \in I} f_i^j F_\alpha(i) \text{ for } \alpha = 1, 2, \dots, N; j \in J$$

These are termed the *transition formulas*. The coordinate of element $i \in I$ (I is the set of scenes) is the barycenter (center of gravity) of the coordinates of the elements $j \in J$ (J is the set of words), with associated masses of value given by the coordinates of f_j^i of the profile f_j^i . (The scene-normalized scene vector of word occurrences is denoted f_j^i , and the word-normalized word vector of scene occurrences is denoted f_i^j .) The α th eigenvalue is λ_α , and there are N axes in all. The transition formulas are all to within the $\lambda_\alpha^{-\frac{1}{2}}$ constant. F and G are the factor projections or coordinates.

The barycenter formulas are very important. From them, we know that a scene is located at a weighted average of all words, given the frequency of occurrence properties of words for that scene. Similarly, we know that a word

is located at a weighted average of all scenes, given the frequency of occurrence properties of scenes for that word.

In practice, Figure 1 presents a very useful view of relationships in our scenes \times words data. But it is an approximation to the full dimensional reality. Therefore we prefer to use the full dimensionality Euclidean representation furnished by the factor space.

2.4 Selecting the Most Pertinent Terms

Presenting a result with around 1700 terms (cf. Figure 1) does not lend itself to convenient display. We ask therefore what the most useful – perhaps the most discriminating terms – are. In correspondence analysis both texts and their characterizing terms are projected into the same factor space. So, from the factor coordinates, we can easily find the closest term(s) to a given text. We do this for each of the 50 scenes and find – in the full dimensionality factor space, and so with no approximation involved – the following for the 50 scenes in succession:

“royce” “soon” “coughs” “tape” “building” “makes” “gasps” “shift” “sign”
“forced” “rushes” “city” “feet” “body” “hotel” “ah” “trying” “or” “business”
“shoes” “screaming” “swab” “gun” “were” “rattle” “print” “really” “brass”
“remember” “judge” “any” “latex” “skin” “both” “herself” “believe” “hospital”
“dress” “finger” “minute” “deep” “statement” “minutes” “shh” “match”
“second” “watching” “enters” “ring” “full”

Among these terms, each individually characterizing a scene, in succession, we note the following. “Royce” is a personal name. Terms like “Ah” and “Shh” are present, and reflective of the scenes. We experimented with other alternatives, such as selecting the more important scenes – which, from our procedure, leads also to the more important words – on the basis of totalled high frequency of occurrence. This did not lead to a more attractive word set (e.g. nouns or verbs). We decided to eschew manual selection of terms, in view of our desire to generalize our results. Finally, we decided to limit ourselves to just one nearest neighbor word, for each scene, on the grounds of facilitating interpretation. However nothing restricts the consideration of, for example, 3 or 4 nearest neighbors.

In section 4.2 below, in all cases, we use the first nearest neighbor word of each of the scenes.

2.5 Hierarchical Clustering of Data Sequence

The correspondence analysis, discussed above, gives us an output Euclidean representation, which in turn facilitates defining distances between scenes and between words. The scenes give us a time-varying signal, or time series. Using the full-dimensionality factor space coordinates, we look at Euclidean distance between successive scenes. Equally valid is how we can use Euclidean distance between the most scene-characterizing words, in sequence. In using full dimensionality, there is no loss of information. Rather, we are using an alternative but equivalent view of the data, relative to the input data set.

In treating the television program script as a linear document in this way, we note the interesting counter-posing of hierarchical to linear – the former being the hierarchy ranging over word, sentence or line, paragraph, and so on. This is discussed in Hearst [1994] and extensively in Chafe [1979]. Perhaps counter-intuitively our linear sequence of script scenes will actually be hierarchically clustered!

We require that the hierarchic clustering should be strictly on the basis of agglomerating or merging clusters that are in (scene ordered) sequence. In accordance with the timeline of scenes (i.e. from 1 to 50 in the case of this particular CSI program introduced in section 2.3), we impose a temporal or sequence-related adjacency constraint on the agglomerative clustering.

The sequence-related adjacency requirement can be viewed from a different perspective. The issue to be considered is whether an agglomerative clustering method would give rise to an inversion, i.e., a later agglomeration in the sequence of agglomerations would have an associated criterion value that is less than the previous criterion value. It is not hard to appreciate that our desire to have gradations of distance represented by the dendrogram would be negated, and severely so, by such absence of the monotonicity of criterion value, which amounts to a contradiction in interpretation of the dendrogram.

It is shown in Murtagh [1985] that just two approaches are feasible. What is involved here is sketched out as follows. Contiguity-constrained single link hierarchical clustering is simultaneously hierarchical clustering on the spanning tree graph. This is easy to implement (inefficiently): just fix an infinite (or very

large) distance between non-contiguous pairs and proceed to use single link hierarchical clustering. The complete link method, with the constraint that at least one member of each of the two clusters to be agglomerated be continuous, is guaranteed not to give rise to inversions. The $O(n^2)$ time, $O(n^2)$ space algorithm for the complete link method, based on the nearest neighbor chain [Murtagh 1985], is easily modified to include an additional testing of contiguity whenever a linkage in the nearest neighbor chain is created. In this work we use the contiguity-constrained (or sequence-respecting) complete link agglomerative criterion, in view of the well-balanced hierarchies typically produced [Murtagh 1984b].

Figure 2 shows the very similar hierarchies yielded by the best-characterizing sequence of 50 words, and the sequence of 50 scenes. We are more interested in inferring an ontology from the former. However we note the inherently very close relationship with the latter.

We will return later to the practical use of such a hierarchy as is shown in Figure 2.

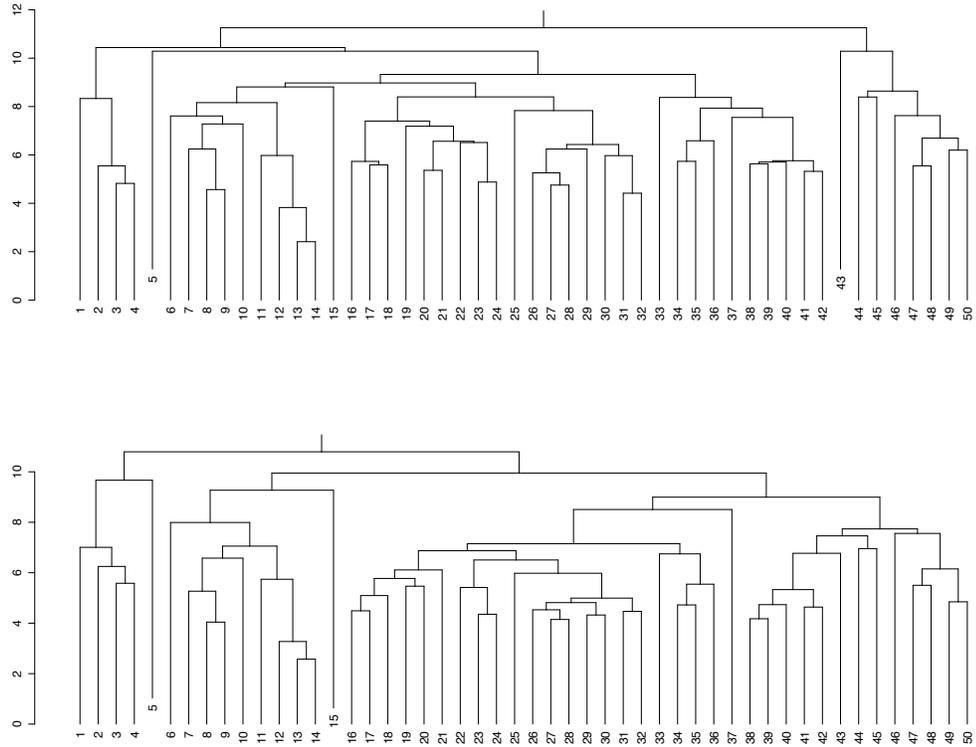


Figure 2: Hierarchical clustering of (top) succession of 50 scene-characterizing words – see subsection 2.4; and (bottom) succession of 50 scenes. In both cases, the complete link agglomerative clustering criterion rigorously respects the given sequence. Since, by design, there is a very good one-to-one mapping of the cloud of 50 words, and the cloud of 50 scenes, both in the same 49-dimensional Euclidean embedding, it follows that the hierarchies are similar.

3 From Hierarchical Clustering to a Hierarchy of Concepts

3.1 A Formal Approach: Displaying a Hierarchical Clustering as an Oriented Tree

We have noted in the Introduction how a hierarchical clustering may be the starting point for creating a concept hierarchy, but the two representations differ. In this section we show how we can move from an embedded set of clusters, to an oriented tree. Orientation in the latter case aims at expressing subsumption.

Consider the dendrogram shown in Figure 3, which represents an embedded set of clusters relating to the 8 terms.

We took Aristotle's *Categories*, which consisted of 14,483 individual words. We broke the text into 24 files, in order to base the textual analysis on the sequential properties of the argument developed. In these 24 files there were 1269 unique words. We selected 66 nouns of particular interest. A sample (with frequencies of occurrence): man (104), contrary (72), same (71), subject (60), substance (58), ... No stemming or other preprocessing was applied. For the hierarchical clustering, we further restricted the set of nouns to just 8. (These will be seen in the figures to be discussed below.) The data array was *doubled* [Murtagh 2005] to produce an 8×48 array, which with removing 0-valued text segments (since, in one text segment, none of our selected 8 nouns

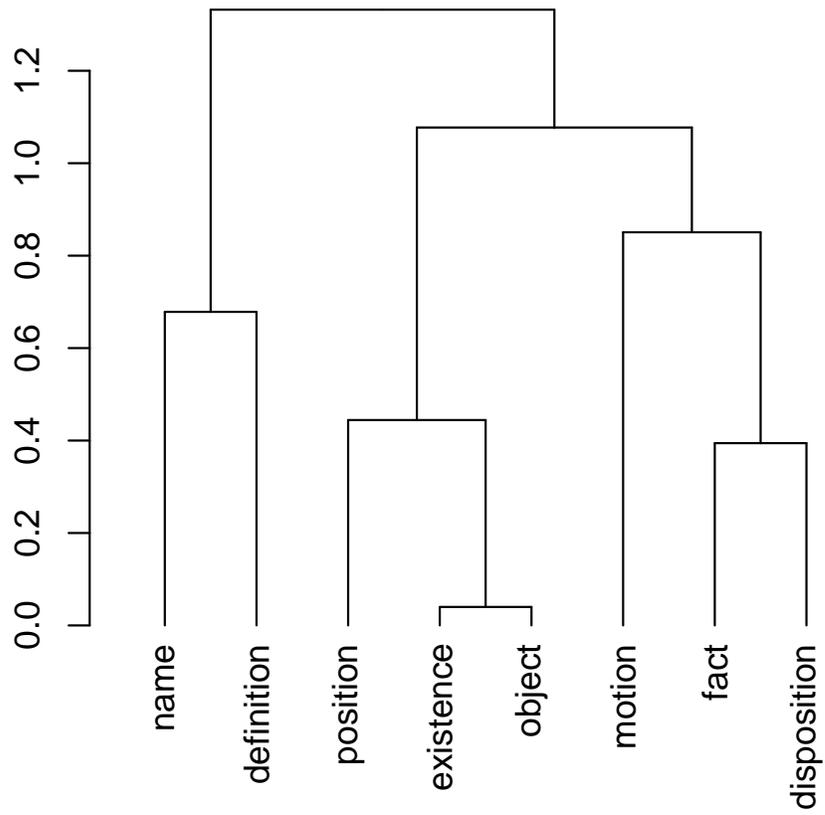


Figure 3: Hierarchical clustering of 8 terms. Data on which this was based: frequencies of occurrence of 66 nouns in 24 successive, non-overlapping segments of Aristotle's *Categories*.

appeared) gave an 8×46 array, thereby enforcing equal weighting of (equal masses for) the nouns used. The spaces of the 8 nouns, and of the 23 text segments (together with the complements of the 23 text segments, on account of the data doubling) are characterized at the start of the correspondence analysis in terms of their frequencies of occurrence, on which the χ^2 metric is used. The correspondence analysis then “euclideanizes” both nouns and text segments. We used the 7-dimensional (corresponding to the number of non-zero eigenvalues found) Euclidean embedding, furnished by the projections onto the factors. A hierarchical clustering of the 8 nouns, characterized by their 7-dimensional (Euclidean) factor projections, was carried out: Figure 3. The Ward minimum variance agglomerative criterion was used, with equal weighting of the 8 nouns.

Figure 4 shows a canonical representation of the dendrogram in Figure 3. Both trees are isomorphic to one another. Figure 4 is shown such that the sequence of agglomerations is portrayed from left to right (and of course from bottom to top). We say that Figure 4 is a canonical representation of the dendrogram, implying that Figure 3 is not in canonical form. In Figure 5, the canonical representation has its non-terminal nodes labeled.

Next, Figure 6 shows a superimposed oriented binary rooted tree, on $n - 1$ nodes, which is isomorphic to the dendrogram on n terminal nodes. This oriented binary tree is an inorder traversal of the dendrogram. Sibson’s [1973] “packed representation” of a dendrogram uses just such an oriented binary rooted tree, in order to define a permutation representation of the dendrogram. From our example, the packed representation permutation can be read off as

(13625748): for any terminal node indexed by i , with the exception of the rightmost which will always be n , define $p(i)$ as the rank at which the terminal node is first united with some terminal node to its right. Discussion of combinatorial properties of dendrograms, as related to such oriented binary rooted trees, and associated down-up and up-down permutations, can be found in Murtagh [1984a].

Finally, in Figure 7, we “promote” terminal node labels to the nodes of the oriented tree. We will use exactly the procedure used above for defining a permutation representation of the oriented tree. First the left terminal label is promoted to its non-terminal node. Next, the right terminal label is promoted as far up the tree as is necessary in order to find an unlabeled non-terminal node. This procedure is carried out for all non-terminal labels, working in sequence from left to right (i.e., consistent with our canonical representation of the dendrogram). The rightmost label is not shown: it is at an arbitrary location in the upper right hand side, with a tree arc oriented towards the top non-terminal node of the dendrogram, now labeled as “motion”.

In this section, we have specified a consistent procedure for labeling the nodes of an oriented tree, starting from the labels associated with the terminal nodes of a dendrogram. We start therefore with embedded clusters, and end up with terms and directed links between these terms. There is some non-uniqueness: any two labels associated with terminal nodes that are left and right child nodes of one non-terminal node can be interchanged. This clearly leads to a different label promotion outcome.

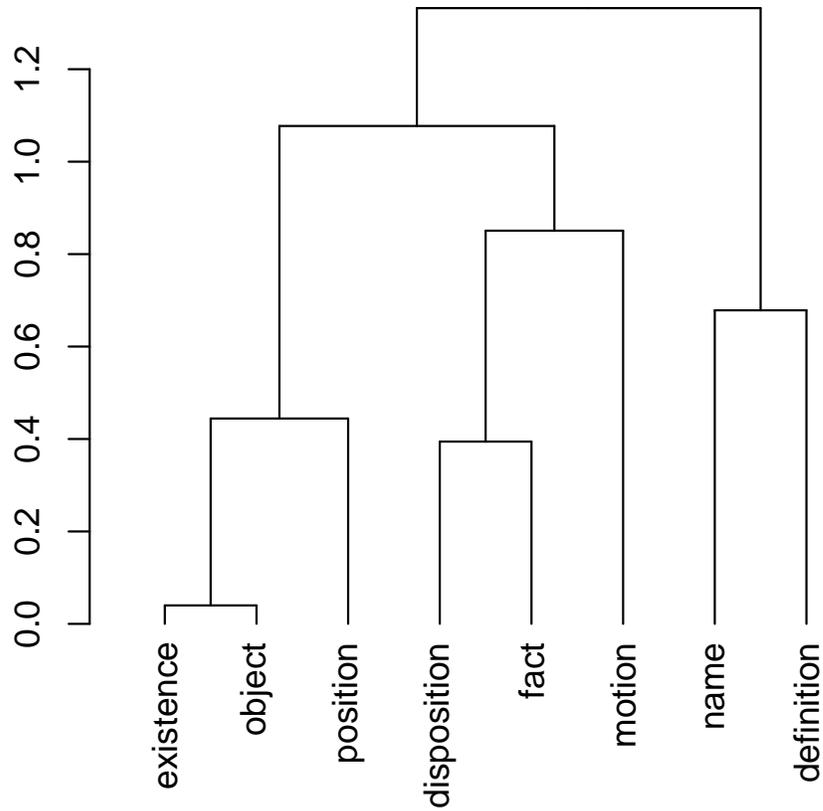


Figure 4: Dendrogram on 8 terms, isomorphic to the previous figure, Figure 3, but now with successively *later* agglomerations always represented by *right* child node. Apart from the labels of the initial pairwise agglomerations, this is otherwise a unique representation of the dendrogram (hence: “existence” and “object” can be interchanged; so can “disposition” and “fact”; and finally “name” and “definition”). In the discussion we refer to this representation, with later agglomerations always parked to the right, as our canonical representation of the dendrogram.

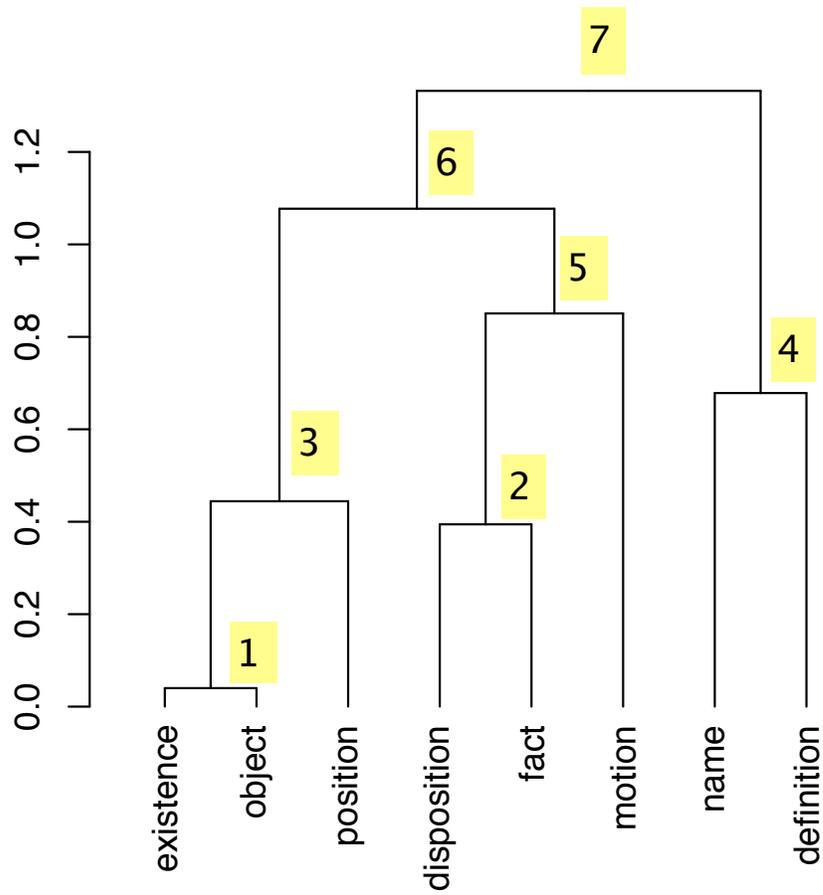


Figure 5: Dendrogram on 8 terms, as previous figure, Figure 4, with non-terminal nodes numbered in sequence. These will form the nodes of the oriented binary tree. We may consider one further node for completeness, 8 or ∞ , located at an arbitrary location in the upper right.

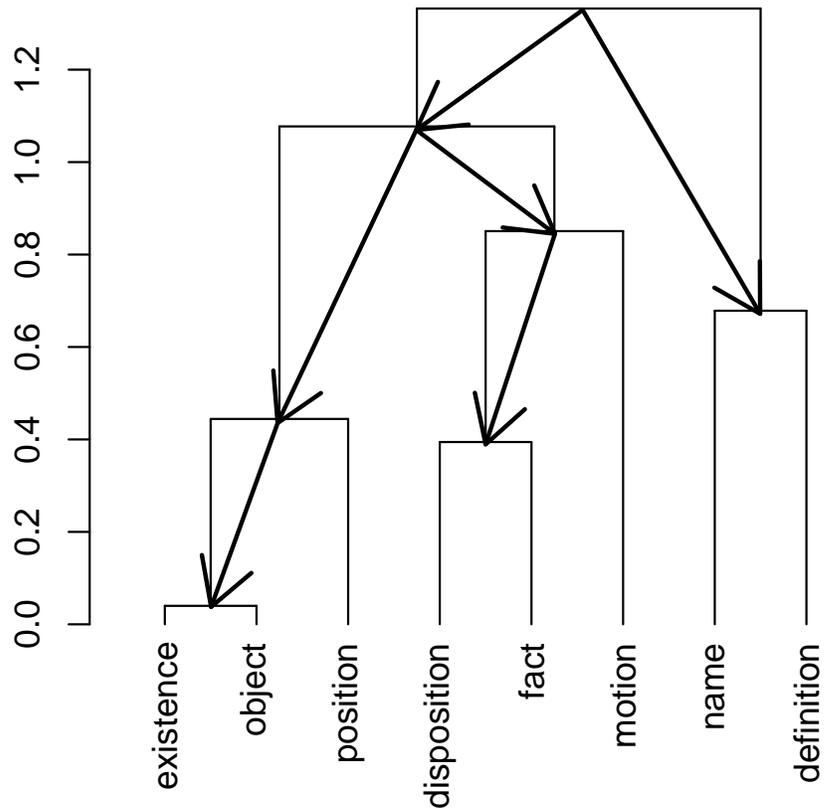


Figure 6: Oriented binary tree is superimposed on the dendrogram. The node at the arbitrary upper right location is not shown. The oriented binary tree defines an inorder or depth-first tree traversal.

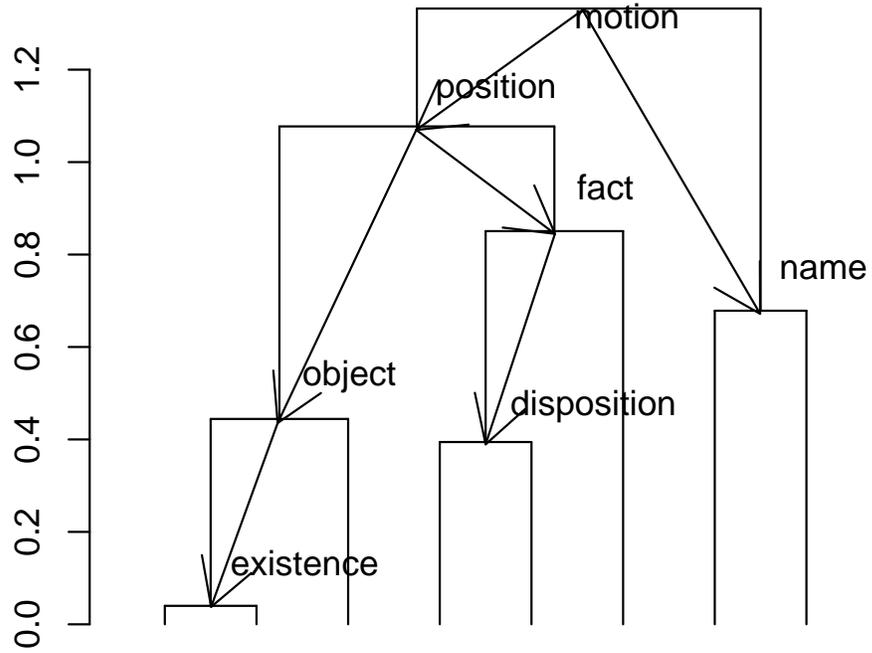


Figure 7: The oriented binary tree with labeled nodes, using the procedure discussed in the text. The label “definition” is not shown and is to be located in the upper right, with an oriented tree arc directed at “motion”.

Our promotion procedure was motivated by the permutation representation of an oriented binary tree, as described above. Here too we do not claim uniqueness of permutation representation. But we do claim optimality in the sense of parsimony, and well-definedness.

In the case of a multiway tree with very few distinct levels, the promotion procedure becomes very simple, but continues to be non-unique.

3.2 A New Approach to Deriving a Concept Hierarchy from a Dendrogram

In the previous subsection, we discussed an algorithm which takes a hierarchical clustering, and hence a dendrogram, into an inorder tree traversal, and hence a permutation of the set of terms used. We now look more closely at implementing this in practice.

The triangle properties between triplets of points, or data objects, are fundamental to ultrametricity and hence to tree representation [Murtagh 2004]. A dendrogram, representing a hierarchical clustering, allows us to read off, for all triplets of points, either (i) isosceles triangles, with small base, or (ii) equilateral triangles, and (iii) no other triangle configuration. The reason for the last condition is simply that non-isosceles, or isosceles with large base, triangles are incompatible with the ultrametric, or tree, metric.

We will leave aside for the present the equilateral triangle case. Firstly, it implies that all 3 points are *ex aequo* in the same cluster. Secondly, therefore we will treat them altogether as a concept cluster. Thirdly, the equilateral case

does not arise in the example we will now explore.

In Figure 8, cluster number 3 indicates the following isosceles triangle with small base: ((existence, object) position). Our notation is: ((x, y) z), such that triplet x, y, z has small base x, y, and the side lengths x, z and y, z are equal. This is necessarily implied by relationships represented in Figure 8. So, motivated by this triangle view of the cluster number 3 part of the dendrogram we will promote “position” to the cluster number 3 node.

Similarly we will promote “motion” to the cluster number 5 node.

Note the consistency of our perspective on the cluster number 3 and 5 nodes relative to how the associated terms here form an isosceles triangle with small base.

We will straight away generalize this definition. In any case of a node in the form of nodes 3 or 5, where we have a 2-term left subtree, and a 1-term right subtree, where left and right are necessarily labeled in this way given the canonical representation of the dendrogram, then: *the left subtree is dominated by the right subtree.*

We will next look at cluster number 6 (remaining with Figure 8). As always for such trees, the node corresponding to this cluster has two subtrees, one to the left (here: 3) and one to the right (here: 5). Since our dendrogram is in canonical form, any such node has a subtree with smallest non-terminal node level to the left; and the subtree which was more recently formed in the sequence of agglomerations to the right. Based on either or both of these criteria which serve to define what are the left and right subtrees we define the ordering

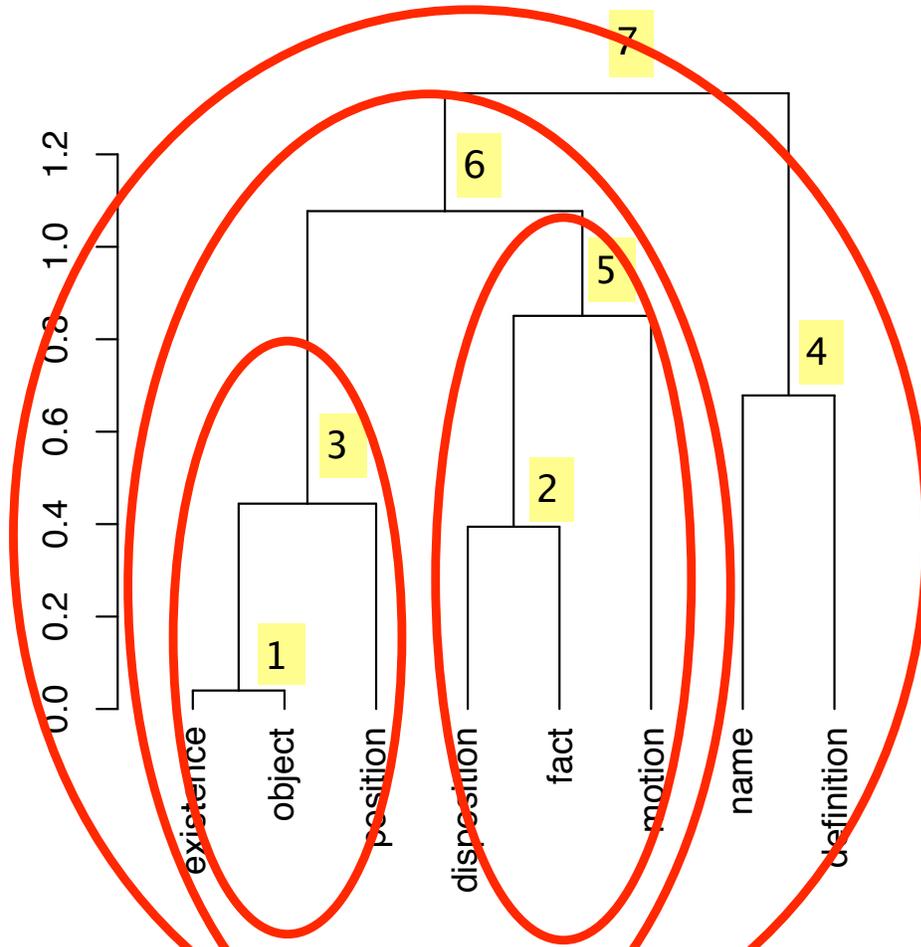


Figure 8: Dendrogram as shown in Figures 4 and 5, with clusters indicated by ellipses. Shown here are ellipses covering the clusters at nodes 7, 6, 5, and 3.

relationship: *the left subtree is dominated by the right subtree.*

Figure 9 summarizes the concept relations that we can derive in a similar way from any dendrogram.

4 Applications

4.1 Successive Text Segments

In this section we present a demonstrator, based on successive text segments of a given text.

Figure 10 indicates how the concept relations, shown in Figure 9, are to be used.

Firstly the term set is summarized, using our selection of terms. Scaling to large data sets is addressed in this way.

Secondly, in our interactive implementation (web address: thames.cs.rhul.ac.uk/~dimitri/textmap), we allow the terms shown to continually move in a limited way, to get around the occlusion problem, and we also allow magnification of the display area for this same reason.

Thirdly, terms other than those shown are highlighted when a cursor is passed over them.

Next, double clicking on any term gives a ranked list of text segment names, ordered by frequency of occurrence by this term. Clicking on the text segment gives the actual text at the bottom of the display area.

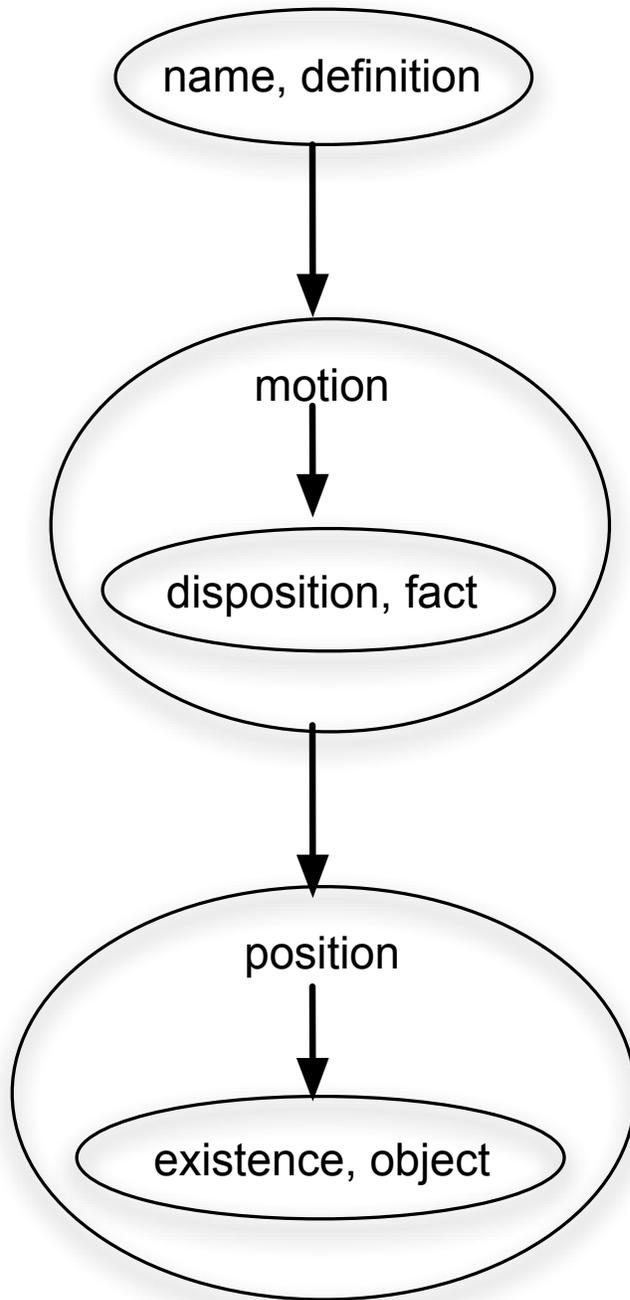


Figure 9: Concept relationships, ordered by dominance, derived from the dendrogram in Figure 8.

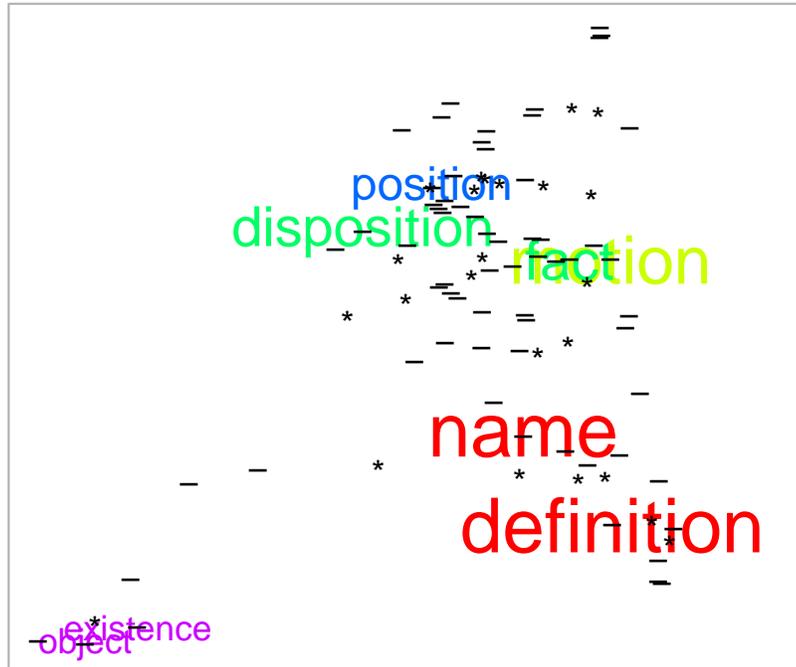


Figure 10: The relationships displayed in Figure 9 are shown in decreasing size (and in rainbow colors, from red), with other terms (in all, 66) displayed with a dash, and all text segments (in all, 24) represented by an asterisk. The principal factor plane of a correspondence analysis (based on the 24 text segments \times 66 terms frequencies of occurrence) output is used.

4.2 Television Transcripts

We now return to the use of this principle of cluster dominance applied to the word dendrogram shown in Figure 2. This is from the CSI 1-01 (program 1 in series 1) script. We find the succession of hierarchically ordered clusters, for just the top part of the dendrogram (to facilitate interpretation) – stopping at the partition with 5 clusters – to be:

Dominant cluster:

“makes” “gasps” “shift” “sign” “forced” “rushes” “city” “feet” “body” “hotel”
“ah” “trying” “or” “business” “shoes” “screaming” “swab” “gun” “were” “rat-
tle” “print” “really” “brass” “remember” “judge” “any” “latex” “skin” “both”
“herself” “believe” “hospital” “dress” “finger” “minute” “deep” “statement”

Next, *ex aequo*, are the clusters:

“minutes”

and:

“shh” “match” “second” “watching” “enters” “ring” “full”

Finally, *ex aequo*, are the following two clusters:

“royce” “soon” “coughs” “tape”

and:

“building”

We started with terms that most closely expressed and characterized the scenes, and indeed that discriminated between the scenes. Using a hierarchical



CSI 1-01 Pilot

arms ass **bathroom** checks clippings **crime** deceased discoloration doll doorway
drives dusting examines fires **flashback** follicles forehead glances god gonna
grabs gurneys **hallway** homicide jar kit kneels lab latex leans **love** nail nods **okay**
opens picks prints **pulls** recorder robbery screams
shadowing **shake** sheets **shuts** sighs sir **sits** smiles spray stares **stops** straightens
suicide **swabs** toe toenail toilet trick underwear victim **walks** wallet yeah

Figure 11: A scriptcloud, showing 64 tags, based on frequent words retained following application of a stoplist. Produced by Scriptcloud, www.scriptcloud.com

clustering that is constrained by the temporal or linear ordering of scenes, we derived a structuring of the words. All aspects of this are based on a Euclidean embedding, which also directly addresses the normalization of the original statistical (i.e. frequency of occurrence) data, both in relation to words and in relation to scenes. Furthermore all aspects of what we have done is automated and does not require any user setting of thresholds or manual selections.

The tag clouds of Figures 11 and 12 use frequency information related to word occurrence, and order words alphabetically. There is nothing in such tag clouds that takes into account the sequential order of the original text. The number of words is set by the user.

CSI101 - Pilot

royce soon coughs tape building makes gasps shift sign forced rushes city feet
body hotel ah trying or business shoes screaming
swab gun were rattle print really brass remember judge any
latex skin both herself believe hospital dress finger minute deep statement minutes shh match
second watching enters ring full

CSI102 - Cool Change

jackpot shakes night suicide word brass want
bringing somebody statement interview intercut stuff
sidewalk money can minute ear grabs sir stay coffee little present officer until leans eyes
watch doubt enough **fibers sees key question sits home**

CSI103 - Crate 'n Burial

listened lamp things under bedroom rag next. heads true screams **heat outside tracks**
minutes their move called accident clear words clamping **prints** amazing
person another moore saying sudden unconscious lot cover tips handcuffs black laura cards watching
slightly

Figure 13: Our tag cloud, where each word is a best characterization of a scene.

The sequence of words corresponds to the sequences of scenes (so the number of words displayed equals the number of scenes). The word font size is proportional to dominance in the hierarchy. Shown are, from top to bottom, three television program scripts.

Our tag cloud in Figure 13 orders words by scene, since the words are – in the sense described in section 2.4 – the best characterizations of the scenes. Thus the most important structuring of the original text, viz. the sequence of scenes, is respected by this output display. The number of words is the same as the number of scenes. We used just the top part of the hierarchy, since we wanted a small number of discrete font sizes to represent groups of words. The number of such groups of words was set as 7 in all cases of analysis of CSI data. (We experimented with more and with less, and present here the case of 7 clusters.) For m words, this corresponds to cutting the dendrogram at level $m = 7$. The 7 successive clusters can be read off, using font size. Our tag cloud, through font size, takes into account dominance as defined by the hierarchical clustering. This is used here to indicate how some terms are to be taken as at a higher level relative to others.

Figure 13 also displays programs two and three of the first series of CSI. Whereas the first of these programs has 50 scenes and 1679 unique words; the second program has 37 scenes and 1343 words; and the third program has 38 scenes and 1413 words.

The dominance relationship, represented as font size, is approximate, for display convenience. Font sizes (integers between 1 and 7 in deprecated HTML) below a default 3 are uncomfortably small, and at 7 too large, so we use a range between 2 and 6, ignoring mapping onto the same value.

To compare these results with other program scripts, we can avail of other data available at TWIZ [2007]. (Seasons 1 through 4 of CSI have each 23

programs; season 5 has 25 programs; seasons 6 and 7 have 24 programs. Season 8 scripts are listed by name only at the time of writing.) From season 4 onwards, the seasons are distinguished by their usual descriptive metadata, but not by a “Scene number” label: since this makes reproducibility of our results a little complicated, we will use some further scripts from the third season. We use the following:

- Third series, 21st program, “Forever”, originally aired on CBS on 1 May 2003, 39 scenes. (In all, 1584 unique words were used.)
- Third series, 22nd program, “Play with Fire”, originally aired on CBS on 8 May 2003, 40 scenes. (In all, 1579 words were used.)
- Third series, 23rd and last program of the series, “Inside the Box”, originally aired on CBS on 15 May 2003, 49 scenes. (In all, 1445 words were used.)

Figure 14 shows the resulting tag clouds. Some remarks on this figure follow. “NV” (CSI322) is part of a much used identifier of personages in a Nevada Correctional Facility in scene 22. Also in CSI322, our punctuation handling has left “doesn” and stripped the remainder.

In these visualizations we see important words; the sequence of important words accompanying the sequence of scenes; sequence-respecting clusters of words; dominance relationships between words; both a view of the temporal and linear text being analyzed, and also various aspects of the hierarchical structure of the text. Properties such as these, seen in the display, can be used

CSI321 - Forever

seat food first nothing stuck ceiling id bindle usda kid pill evidence pinpoints harper bodies carrying
confer stitching sitting daughter vic greater uterus blonde give table tax banks best **area car** missing warrant
pills going one smuggle doing rhone

CSI322 - Play with Fire

enough scene put checks container car neck motel former match your bite
place steps head are never attorney about cases
number nv groupie talk puts open passes doesn't heroin hot could green officer ready director those lawyer
everything eyes too

CSI323 - Inside the Box

little guns glass open woman grabs weapon cut police past weren't leads muffled **check snaps**
part tells table hood lockboxes used could call casino home enough left truth office shows
murdock braun means sure wish then cuts address cronies robbery man tested hands scarf murder care row
vivian arm

Figure 14: Tag clouds for programs (episodes) 21, 22 and 23 – the final ones –
in CSI Series 3. See Figure 13 for other details.

to distinguish one episode (program) from another.

One of the most appealing aspects of our approach is that all phases of the processing are automated, and there are no user-set parameters or other user intervention required.

We checked that the pertinent words found to characterize each scene were unique. This was always the case. If the same word were found to be closest to two scenes we could of course use any such multiple occurrences of words. If separated in the timeline or sequence of words, then they would of necessity be separated also in the dendrogram.

5 Conclusions

With reference to Chafe [1979] we used the sequence of text segments representing scenes in the script (or the segments derived from the Aristotle text), and we also used internal hierarchical structure. Sequence, clustering, hierarchy are all presented comfortably and clearly in our tag cloud display.

Computationally, most of the processing is of linear time. The contiguity-constrained hierarchical clustering, for convenience in implementation, used a quadratic time algorithm. Using a nearest neighbor chain algorithm, and storage and dynamic updating of the pairwise dissimilarities, we can envisage a linear time algorithm, benefiting from the total order of the sequence of agglomerands. We conclude that our analysis pipeline has excellent scalability properties.

Our approach is automated, without recourse to user parameters, or user

choice of chained tasks. A framework enveloping input data and delivered (potentially interactive and responsive) display is provided by mappings between χ^2 and Euclidean metrics, and ultrametric topologies.

We associate closely both local hierarchic structure in the data, expressed via ultrametric or strong triangular inequality properties, and global, fitted hierarchical structure. Reasonable strength of the former properties justify the latter. We have also clearly formulated and proposed a solution for mapping a hierarchy from an object set I , as furnished by a hierarchical clustering, to a concept set furnished by a subset of the power set of I , 2^I .

Our global objective [Murtagh et al. 2003] is to support self-description of data as a basis for visually-based interactive and responsive querying of, retrieval from, and navigation through, data and information stores. The focus of this article has been on the building of ontologies, to aid operation and deployment of, in this sense, active information.

Acknowledgements

This work was started in the EU Sixth Framework project, “WS-Talk, Web services communicating in the language of their community”, 2004–2006. Pedro Contreras and Dimitri Zervas contributed to this work. The Textmap demonstrator was developed by Dimitri Zervas, who also wrote the programs for generating word lists and cross-tabulation tables. The hierarchical clustering and correspondence analysis programs were written by Fionn Murtagh. Analysis and exploration was carried out in the R environment. Extensive discussions

were held with Adam Ganz.

References

- ABOU ASSALI, A. and ZANGHI, H. (2006). Automated metadata hierarchy derivation. In *Proc. ICTTA '06, Information and Communication Technologies*, IEEE, pp. 505–510.
- AHMAD, K., GILLAM, L. and TOSTEVIN, L. (1999). Weirdness indexing for logical document extrapolation and retrieval (WILDER). In E.M. Voorhees and D.K. Harman, Eds., *TREC-8: The 8th Text Retrieval Conference*, NIST, pp. 717–724.
- AHMAD, K. and GILLAM, L. (2005). Automatic ontology extraction from unstructured texts. In *On the Move to Meaningful Internet Systems – OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005*, Lecture Notes in Computer Science 3761, R. Meersman and Z. Tari, Eds., Springer, pp. 1330–1346.
- BENZÉCRI, J.P. (1979). *L'Analyse des Données, Tome I Taxinomie, Tome II Correspondances*, 2nd ed. (Dunod, Paris).
- BUITELAAR, P., CIMIANO, P., and MAGNINI, B. (2005). Ontology learning from text: An overview. In: P. Buitelaar, P. Cimiano, B. Magnini, Eds., *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence and Applications Series, Vol. 123,

IOS Press.

CHAFE, W.L. (1979). The flow of thought and the flow of language. In T. Givón, Ed., *Syntax and Semantics: Discourse and Syntax*, Vol. 12, Academic Press, 159–181.

CHAN, S.W.K. (2004). Extraction of salient textual patterns: synergy between lexical cohesion and contextual coherence, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 34 (2), 205-218.

CHAN, S.W.K. (2006). Beyond keyword and cue-phrase matching: A sentence-based abstraction technique for information extraction, *Decision Support Systems*, 42 (2), 759–777.

CHUANG SHUI-LUNG and CHIEN LEE-FENG (2005). Taxonomy generation for text segments: a practical web-based approach, *ACM Transactions on Information Systems*, 23, 363–396.

CIMIANO, P., HOTHO, A., and STAAB, S. (2005). Learning concept hierarchies from text corpora using Formal Concept Analysis, *Journal of Artificial Intelligence Research*, 24, 305–339.

DAVEY, B.A. and PRIESTLEY, H.A. (2002). *Introduction to Lattices and Order*, 2nd edn. (Cambridge University Press, Cambridge).

DE SOETE, G. (1986). A least squares algorithm for fitting an ultrametric tree to a dissimilarity matrix. *Pattern Recognition Letters*, 2, 133–137.

- DENNY, M. (2004). Ontology tools survey, Revisited, at <http://www.xml.com/pub/a/2004/07/14/onto.html>, July 2004.
- DOYLE, L.B. (1961). Semantic road maps for literature searches. *Journal of the ACM*, 8, 553–578.
- GANESAN, P., GARCIA-MOLINA, H. and WIDOM, J. (2003). Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*. 21, 64–93.
- GILLAM, L., TARIQ, M. and AHMAD, K. (2005). Terminology and the construction of ontology. *Terminology*, 11, 55–81.
- GLADWELL, M. (2006). The formula: What if you build a machine to predict hit movies? *The New Yorker*, October 16, 2006.
http://www.newyorker.com/archive/2006/10/16/061016fa_fact6
- GÓMEZ-PÉREZ, A., FERNÁNDEZ-LÓPEZ, M. and CORCHO, O. (2004). *Ontological Engineering (with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web)* (Springer, Berlin).
- GRUBER, T. (2001). What is an ontology?, <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>, Sept. 2001.
- HEARST, M. (1994). Multi-paragraph segmentation of expository text, *Annual Meeting of the ACL, Proc. 32nd Annual Meeting of Association for Computational Linguistics*, 9–16.

- DE LA HIGUERA, C. and DANIEL-VATONNE, M.-C. (1996). On sets of terms: a study of a generalisation relation and of its algorithmic properties. *Fundamenta Informaticae*, 25, 99–121.
- JANOWITZ, M.F. (2005). Cluster analysis based on abstract posets, preprint. Also presentation, ENST de Bretagne, 30 Oct. 2004; DIMACS, 9 Mar. 2005; and SFC, 31 May 2005.
- KNAPP, E.W. (1988). Equivalence of dynamics in ultrametric and hierarchical spaces. *Physical Review B*, 38, 2664–2668.
- KOHONEN, T., KASKI, S., LASKI, K., SALOJÄRVI, J., HONKELA, J., PAATERO, V. and SAARELA, A. (2000). Self-organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11, 574–585.
- LEE, C.-S., KAO, Y.-F., Kuo, Y.-H. and WANG, M.-H. (2007). Automated ontology construction for unstructured text documents. *Data and Knowledge Engineering*, 60, 547–566.
- MAEDCHE, A. (2006). *Ontology Learning for the Semantic Web* (Kluwer, Dordrecht).
- McKIE, S. (2007). Scriptcloud: content clouds for screenplays. Report. See also: Screenplay content clouds, www.scriptcloud.com
- MURTAGH, F. (1984a). Counting dendrograms: a survey. *Discrete Applied Mathematics*, 7, 191–199.

- MURTAGH, F. (1984b). Structures of hierarchic clusterings: implications for information retrieval and for multivariate data analysis. *Information Processing and Management*, 20, 611–617.
- MURTAGH, F. (1985). *Multidimensional Clustering Algorithms* (Physica-Verlag, Würzburg, Vienna).
- MURTAGH, F. (2004). On ultrametricity, data coding, and computation, *Journal of Classification*, 21, 167–184.
- MURTAGH, F. (2005). *Correspondence Analysis and Data Coding with R and Java* (Chapman & Hall/CRC, Boca Raton).
- MURTAGH, F. (2006a). Ultrametricity in data: identifying and exploiting local and global hierarchical structure, arXiv:math.ST/0605555v1 19 May 2006.
- MURTAGH, F. (2006b). Symbolic dynamics in text: application to automated construction of concept hierarchies, *Festschrift for Edwin Diday* (Springer, Heidelberg) in press. Available at: www.cs.rhul.ac.uk/home/fionn/papers
- MURTAGH, F. (2006c). Visual user interfaces, interactive maps, references, theses, <http://astro.u-strasbg.fr/~fmurtagh/inform>
- MURTAGH, F. (2007). The Haar wavelet transform of a dendrogram, *Journal of Classification*, 24, 3–32.
- MURTAGH, F., TASKAYA, T., CONTRERAS, P., MOTHE, J. and ENGLMEIER, K. (2003). Interactive visual user interfaces: a survey, *Ar-*

tificial Intelligence Review, 19, 263–283.

MURTAGH, F., DOWNS, G. and CONTRERAS, P. (2007). Hierarchical clustering of massive, high dimensional data sets by exploiting ultrametric embedding, *SIAM Journal on Scientific Computing*, in press.

NAVIGLI, R. and VELARDI, P. (2004). Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30, 151–179.

SCHUTZ, A. and BUITELAAR, P. (2005). RelExt: A tool for relation extraction in ontology extension, In *Proc. of the 4th International Semantic Web Conference*, Galway.

SIBSON, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method, *Computer Journal*, 16, 30–34.

SPÄRCK JONES, K. (1971). *Automatic Keyword Classification for Information Retrieval* (Butterworths, London).

TUCKER, R.I. and SPÄRCK JONES, K. (2005). Between shallow and deep: an experiment in automatic summarising, University of Cambridge Technical Report No. 632, UCAM-CL-TR-632, pp. 34.

TWIZ TV, CSI Transcripts, Seasons 1 to 8, www.twiztv.com/scripts/csi

VELARDI, P., CUCCHIARELLI, A. and PÉTIT, M. (2007). A taxonomy learning method and its application to characterize a scientific web community, *IEEE Transactions on Knowledge and Data Engineering*, in press.

WACHE, H., VÖGELE, T., VISSER, U., STUCKENSCHMIDT, H., SCHUSTER, G., NEUMANN, H. and HÜBNER, S. (2001). Ontology-based integration of information – a survey of existing approaches. In *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*.

XIAOMENG SU and GULLA, J.A. (2006). An information retrieval approach to ontology mapping, *Data and Knowledge Engineering*, 58, 47–69.