Journal of Machine Learning Research 3 (2002) 747-780 Submitted: 12/01; Published: 12/02

Coupled Clustering: A Method for Detecting Structural Correspondence

Zvika Marx MARXZV@CS.BIU.AC.IL
The Interdisciplinary Center for Neural Computation
The Hebrew University of Jerusalem
Givat-Ram, Jerusalem 91904, Israel
and Department of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel

Ido Dagan DAGAN@CS.BIU.AC.IL
Department of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel

Joachim M. Buhmann JB@INFORMATIK.UNI-BONN.DE
Institut für Informatik III
University of Bonn
Römerstr. 164, D-53117 Bonn, Germany

Eli Shamir SHAMIR@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Givat-Ram, Jerusalem 91904, Israel

Editors: Carla E. Brodley and Andrea Danyluk

Abstract

This paper proposes a new paradigm and a computational framework for revealing equivalencies (analogies) between sub-structures of distinct composite systems that are initially represented by unstructured data sets. For this purpose, we introduce and investigate a variant of traditional data clustering, termed *coupled clustering*, which outputs a configuration of corresponding subsets of two such representative sets. We apply our method to synthetic as well as textual data. Its achievements in detecting topical correspondences between textual corpora are evaluated through comparison to performance of human experts.

Keywords: Unsupervised learning, Clustering, Structure mapping, Data mining in texts, Natural language processing

1. Introduction

Unsupervised learning methods aim at the analysis of data, based on patterns within the data itself while no supplementary directions are provided. Two extensively studied unsupervised tasks are: (a) assessing similarity between object pairs, typically quantified by a single value denoting an overall similarity level, and (b) detecting, through various techniques, the detailed structure of individual composite objects.

The common approach to similarity assessment has been to examine feature-based vectorial representations of each object in order to calculate some distance or proximity measure between object pairs (see Subsection 2.2 below for examples). A natural extension to this task would be to analyze in detail *what* makes two composite

©2002 Zvika Marx, Ido Dagan, Joachim M. Buhmann and Eli Shamir

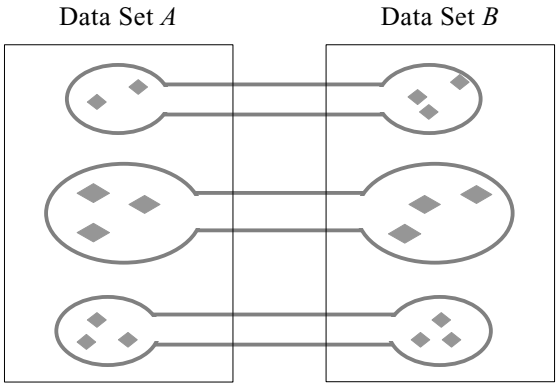

Figure 1: The coupled clustering framework. The diamonds represent elements of the two clustered data sets A and B. Closed contours represent coupled clusters, capturing corresponding sub-structures of the two sets.

objects similar. According to the common view, two given objects are considered similar if they share a relatively large subset of common features, which are identifiable independently of the role they perform within the internal organization of the compared objects. Nonetheless, one might wonder how faithfully a single value (or a list of common features) represents the whole richness and subtlety of what could be conceived as similar. Indeed, cognitive studies make a distinction between surface-level similar appearance and deep structure-based correspondence relationships, such as in analogies and metaphors. Motivated by this conception of structure-related similarity, we introduce in this paper a novel conceptual and algorithmic framework for unsupervised identification of structural correspondences.

Data clustering techniques—relying themselves on vectorial representations or similarities within the clustered elements—impose elementary structure on a given unstructured corpus of data by partitioning it into disjoint clusters (see Subsection 2.1 for more details). The method that we introduce here—*coupled clustering*—extends the standard clustering methods for a setting consisting of a pair of distinct data sets. We study the problem of partitioning these given two sets into corresponding subsets, so that every subset is matched with a counterpart in the other data set. Each pair of matched subsets forms jointly a *coupled cluster*. A resulting configuration of coupled clusters is sketched in Figure 1. A coupled cluster consists of elements that are similar to one another and distinct from elements in other clusters, subject to the context imposed by aligning the clustered data sets with respect to each other. Coupled clustering is intended to reflect comparison-dependent equivalencies rather than overall similarity. It produces interesting results in cases where the two clustered data sets are overall not very similar to each other. In such cases, coupled clustering yields inherently different outcome than standard clustering applied to the union of the two data sets: standard clustering might be inclined to produce clusters that are exclusive to elements of either data set. Coupled clustering, on the other hand, is directed to include representatives from both data sets in every cluster. Coupled clustering is a newly defined computational task of general purpose. Although it is ill posed, similarly to the standard data-clustering problem, it is potentially usable for a variety of applications.

Coupled clustering can potentially be utilized to reveal equivalencies in any type of data that can be represented by unstructured data sets. Representative data sets might contain, for instance, pixels or contours sampled from two image collections, or patterns of physiological measurements collected from distinct populations and so on.

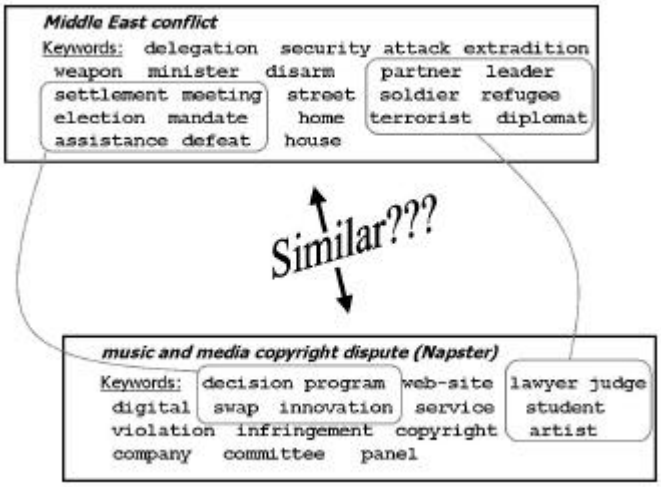

Figure 2: Keyword samples from news articles regarding two conflicts. Examples of coupled clusters, i.e. matched pairs of corresponding keyword subsets, are marked by curved contours.

The current work concentrates on textual data, while the variety of other conceivable applications should be investigated within subsequent projects. Specifically, we apply coupled clustering to pairs of textual sources—document collections or corpora—containing information regarding two distinct topics that are characterized by their own terminology and key-concepts. The target is to identify prominent themes, categories or entities, for which a correspondence can be identified simultaneously within both corpora. The keyword sets in Figure 2, for instance, have been extracted from news articles regarding two conflicts of distinct types: the Middle-East conflict and the dispute over copyright of music and other media types (the "Napster case"). The question of whether, and with relation to which aspects, these two conflicts are similar does not seem amenable to an obvious straightforward analysis. Figure 2, however, demonstrates some non-trivial correspondences that have been identified by our method. For example: the role played within the Middle East conflict by individuals—such as 'soldier', 'refugee', 'diplomat'—has been aligned by our procedure, in this specific comparison, with the role of other individuals—'lawyer', 'student' and 'artist'—in the copyright dispute.

Cognitive research, most notably the *structure mapping theory* (Gentner, 1983), has emphasized the importance of the various modes of similarity assessment for a variety of mental activities. In particular, the role of structural correspondence is crucial in analogical reasoning, abstraction and creative thinking, involving mental maps between complex domains, which might appear unrelated at first glance. The computational mechanisms introduced for implementing the structure mapping theory and related approaches typically require as input data items that are encoded a priori with structured knowledge (see Section 7 for further discussion). Our framework, in distinction, assumes knowledge of abstracted type, namely similarity values pertaining to unstructured data elements, while the structure that underlies the correspondence between the compared domains emerges through the mapping process itself.

Structural equivalencies consistently enlighten various fields of knowledge and scholarship. Historical situations and events, for instance, provide a rich field for the construction of structural analogies. A unique enterprise dates back to Plutarch's "Parallel Lives", in which Greek public figures are paired with Roman counterparts

whose "feature vectors" of life events and actions exhibit structural similarity.[1] In comparison to ancient times, the current era presents growing accessibility to large amounts of unstructured information. Indeed, intensive research takes place, within areas such as information retrieval and data mining, aiming at the needs that evolve in an information-intensive environment. This line of research typically addresses similarity in its surface feature-based mode. A subsequent objective would be to relate and compare separate aggregations of information with one another through structure mapping. In the field of competitive intelligence, for example, one attempts to obtain knowledge of the players in a given industry and to understand the strengths and weaknesses of the competitors (Zanasi, 1998). In many cases, plenty of data—financial reports, white papers and so on—are publicly available for any type of analysis. The ability to map automatically knowledge regarding products, staff or financial policy of one company onto the equivalent information of another company could make a valuable tool for inspection of competing firms. If applied appropriately to such readily available data, structure mapping might turn into a useful approach in future information technology.

The current paper extends earlier versions of the coupled clustering framework that have been presented previously (Marx, Dagan and Buhmann, 2001; Marx and Dagan, 2001). In Section 2, we review the computational methods used within our procedure, namely standard clustering methods and co-occurrence based similarity measures. The coupled clustering method is formally introduced in Section 3. Then, we demonstrate our method's capabilities on synthetic data (Section 4) as well as in detecting equivalencies in textual corpora, including elaboration of the conflict example of Figure 2 and identification of corresponding aspects within various religions (Section 5). Evaluation is conducted through comparison of our program's output with clusters that were constructed manually by experts of comparative studies of religions (Section 6). Thereafter, we compare the coupled clustering method with related research (Section 7). In Section 8, we illustrate how coupled clustering, which is essentially a feature-based method, could be used to detect equivalent relational patterns of the type that have been motivating cognitive theories of structural similarity. The paper ends with conclusions and directions for further research (Section 9).

2. Computational Background

The following two subsections address the computational procedures that are utilized within the coupled-clustering framework. The first subsection, reviewing the data-clustering task, concentrates on the relevant details of the particular approach that we have adapted for our algorithm (Subsection 3.3), by Puzicha, Hofmann and Buhmann (2000). The following subsection reviews methods for calculating similarity values. It exemplifies co-occurrence based techniques of similarity assessment, utilized later for generating input for coupled clustering applied to keyword data sets extracted from textual corpora (Sections 5, 6).

2.1. Cost-based Pairwise Clustering

Data clustering methods provide a basic and widely used tool for detecting structure within initially unstructured data. Here we concentrate on the basic clustering task, namely partitioning the given data set into relatively homogenous and well-separated clusters. Hereinafter, we shall refer to this standard task by the term *standard*

[1] Plutarch's "Lives" can be browsed at **http://classics.mit.edu/Browse/browse-Plutarch.html**.

clustering, to distinguish it from coupled clustering. The notion of data clustering can be extended to include several additional methods, which would not be discussed further here, for instance, methods that output overlapping clusters or hierarchical clustering that recursively partitions the data.

Our motivation for utilizing a clustering mechanism for mapping structure stems from the view of clustering as extracting of meaningful components in the data (Tishby, Pereira and Bialek, 1999). This view is particularly sensible when the clustered data is textual. In this case, clusters of words (Pereira, Tishby and Lee, 1993) or documents (Lee and Seung, 1999; Dhillon and Modha, 2001) can be referred to as explicating prominent semantic categories, topics or themes that are substantial within the analyzed texts.

Clustering often relies on associating data elements with feature vectors. In this case, each cluster can be represented by some averaged (centroid) vectorial extraction (e.g., Pereira, Tishby and Lee, 1993; Slonim and Tishby, 2000b). An alternative approach, *pairwise clustering*, is based on a pre-given measure of similarity (or distance) between the clustered elements, which are not necessarily embeddable within a vector space or even a metric space. Feature vectors, if not used directly for clustering, can still be utilized for defining and calculating a pairwise similarity measure.

The clustering task is considered ill-posed: there is no pre-determined criterion measuring objectively the quality of any given result. However, there are clustering algorithms that integrate the effect of the input—feature vectors or similarity values—through an objective function, or *cost function*, which assigns to any given partitioning of the data (*clustering configuration*) a value denoting its assumed quality.

The clustering method that we use in our work follows a cost-based framework for pairwise clustering recently introduced by Puzicha, Hofmann and Buhmann (2000). They present, analyze and classify a family of clustering cost functions. We review here relevant details of their framework, to be adapted and modified for coupled clustering later (Section 3).

A clustering procedure partitions the elements of a given data set, A, into disjoint subsets, $A_1, \ldots, A_k$. Puzicha et al.'s framework assumes "hard" assignments: every data element is assigned into one and only one of the clusters. Ambiguous, or soft, assignments can be considered advantageous in recording subtleties and ambiguities within lingual data, for example. However, there are reasons to adhere first to hard assignments. It is technically and conceptually simpler and it constructs definite and easily interpretable clustering configurations (Section 7 further addresses this topic). The number of clusters, k, is pre-determined and specified as an input parameter to the clustering algorithm.

A cost criterion guides the search for a suitable clustering configuration. This criterion is realized through a cost function $H(S, M)$ taking the following parameters:

(i) $S = \{s_{aa'}\}_{a,a' \in A}$: a collection of pairwise similarity values[2], each of which pertains to a pair of data elements a and a' in A.

(ii) $M = (A_1, \ldots, A_k)$: a candidate clustering configuration, specifying assignments of all elements into the disjoint clusters (that is $\bigcup A_j = A$ and $A_j \cap A_{j'} = \emptyset$ for every $1 \leq j \neq j' \leq k$).

[2] In their original formulation, Puzicha et al. use distance values (dissimilarities) rather then similarities. Hereinafter, we apply straightforward adaptation to similarity values by adding a minus sign to H. Adhering to the cost minimization principle, this transformation replaces the cost paid for within-cluster dissimilarities with cost saved for within-cluster similarities (alternatively pronounced as "negative cost paid").

The cost function outputs a numeric cost value for the input clustering-configuration M, given the similarity collection S. Thus, various candidate configurations can be compared and the best one, i.e the configuration of lowest cost, is chosen. The main idea, underlying clustering criteria, is the preference of configurations in which similarity of elements within each cluster is generally high and similarity of elements that are not in the same cluster is correspondingly low. This idea is formalized by Puzicha et al. through the *monotonicity* axiom: in a given clustering configuration, locally increasing similarity values, pertaining to elements within the same cluster, cannot decrease the cost assigned to that configuration. Similarly, increasing the similarity level of elements belonging to distinct clusters cannot increase the cost.

Monotonicity is adequate for pairwise data clustering. By introducing further requirements, Puzicha et al. focus on a more confined family of cost functions. The following requirement focuses attention on functions of relatively simple structure. A cost function H fulfills the *additivity* axiom if it can be presented as the cumulative sum of repeated applications of "local" functions referring individually to each pair of data elements. That is:

$$H(S, M) = \sum_{a,a' \in A} \psi^{aa'}(a, a', s_{aa'}, M) , \quad (1)$$

where $\psi^{aa'}$ depends on the two data elements a and a', their similarity value, $s_{aa'}$, and the whole clustering configuration M. An additional axiom, the *permutation invariance* axiom, states that cost should be independent of element and cluster reordering. Combined with the additivity axiom, it implies that a single local function ψ, s.t. $\psi^{aa'} \equiv \psi$ for all $a,a' \in A$, can be assumed.

Two additional invariance requirements aim at stabilizing the cost under simple transformations of the data. First, relative ranking of all clustering configurations should persist under scalar multiplication of the whole similarity ensemble. Assume that all similarity values within a given collection S are multiplied by a positive constant c, and denote the modified collection by cS. Then, H fulfills the *scale invariance* axiom if for every fixed clustering configuration M, the following holds:

$$H(cS, M) = cH(S, M) . \quad (2)$$

Likewise, it is desirable to control the effect of an addition of a constant. Assume that a fixed constant Δ is added to all similarity values in a given collection S, and denote the modified collection by $S^{+\Delta}$. Then, H fulfills the *shift invariance* axiom if for every fixed clustering configuration M, the following holds:

$$H(S^{+\Delta}, M) = H(S, M) + \Phi , \quad (3)$$

where Φ may depend on Δ and on any aspect of the clustered data (typically the data size), but not on the particular configuration M.

As the most consequential criterion, to assure that a given cost function is not subject to local slips, Puzicha et al. suggest a criterion for *robustness*. This criterion ensures that whenever the data is large enough, bounded changes in the similarity values regarding one specific element, $a \in A$, would result in limited effect on the cost. Consequently, the cost assigned to any clustering configuration would not be susceptive to a small number of fluctuations in the similarity data. Formally, denote the size of the data set A by n and let $S^{a+\Delta}$ be the collection obtained by adding Δ to all similarity values in S pertaining to one particular element, $a \in A$. Then H is robust (in the strong sense) if it fulfills

$$\frac{1}{n}\,|\,H(S,M)-H(S^{a+\Delta},M)\,|\xrightarrow[n\to\infty]{}0\,. \tag{4}$$

It turns that among the cost functions examined by Puzicha et al. there is only one function that retains the characterizations given by Equations 1, 2, 3 above, as well as the strong robustness criterion of Equation 4. This function, denoted here as H^0, involves only within-cluster similarity values, i.e. similarity values pertaining to elements within the same cluster. Specifically, H^0 is a weighted sum of the average similarities within the clusters. Denote the sizes of the k clusters $A_1, \ldots, A_k$ by $n_1, \ldots, n_k$ respectively. The average within-cluster similarity for the cluster A_j is then

$$Avg_j = \frac{\sum_{a,a'\in A_j} s_{aa'}}{n_j \times (n_j - 1)}\,. \tag{5}$$

H^0 weights the contribution of each cluster to the cost proportionally to the cluster size:

$$H^0 = -\sum_j n_j Avg_j\,. \tag{6}$$

In Section 3, we modify H^0 to adapt it for the coupled clustering setting.

2.2. Feature-based Similarity Measures

Similarity measures are used within many applications: data mining (Das, Mannila and Ronkainen, 1998), image retrieval (Ortega et al., 1998), document clustering (Dhillon and Modha, 2001), and approximation of syntactic relations (Dagan, Marcus and Markovitch, 1995), to mention just few. The current paper aims at a different approach to similarity of composite objects, more detailed than the conventional single-valued similarity measures. However, as a pre-processing step, preceding the application of the coupled clustering procedure, we calculate similarity values pertaining to the data elements, which are, in our experiments, keywords extracted from textual corpora. The required similarity values can be induced, in principle, in several ways: they could be obtained, for example, through similarity assessments by experts or naive human subjects that were exposed to the relevant data. An alternative way is to calculate similarities from feature vectors representing the data elements.

There are many alternatives, as well, for obtaining appropriate feature-based vectorial representations. The method for this heavily depends, of course, on the specific data under study. In general terms and within textual data in particular, the *context* in which data elements are observed is often used for feature extraction. This approach conforms to the observation that two objects—for example, keywords extracted from a given corpus—are similar if they consistently occur in similar contexts. Thus, a keyword can be represented as a list of values referring to other words co-occurring with it along the text, e.g., the corresponding co-occurrence counts or co-occurrence probabilities. In this representation, each dimension in the resulting feature space corresponds to one of the co-occurring words. The resulting (sparse) vectors, whose entries are co-occurrence counts or probabilities, can underlie distance or similarity calculations.

Numerous studies concerning co-occurrence-based measures have been directed to calculating similarity of words. The scope of co-occurrence ranges from counting occurrences within their specific syntactic context (Lin, 1998) to a sliding window of 40-50 words (Gale, Church and Yarowsky, 1993) or an entire document (Dhillon and Modha, 2001).

A widely used measure of similarity between co-occurrence vectors of words is their cosine, i.e. dot product of the normalized vectors (used e.g., by Dhillon and

Modha, 2001). This measure yields 1 for identical co-occurrences vector (such as the case of self-similarity), and 0 if the vectors are orthogonal, i.e. the two corresponding keywords do not commonly co-occur with any word. The rest of the cases yield values between 0 and 1, in correlation with the degree of overlap of co-occurrences. This measure, similarly to other straightforward measures, is affected by the data sparseness problem: the common use of non-identical words for reference to similar contexts. One strategy for coping with this issue is to project the co-occurrence data into a subspace of lower dimension (LSI: Latent Semantic Indexing, Deerwester et al., 1990; Schutze, 1992; NMF: Non-negative Matrix Factorization, Lee and Seung, 1999).

In our calculations, the same issue is tackled through a simpler approach that does not alter the feature space, but rather puts heavier weights on features that are more informative. The information regarding a data element, x, conveyed through a given feature, w, for which similarity is being measured, is assessed through the following term:[3]

$$I(x,w) = \log_2^+ \frac{p(x \mid w)}{p(x)} \ , \tag{7}$$

where, p denotes conditional and unconditional occurrence probabilities and the '+' sign indicates that 0 is returned whenever the $\log_2$ function produces negative value.

Dagan, Marcus and Markovitch (1995) base their similarity measure on this term:

$$sim_{DMM}(x_1,x_2) = \frac{\sum_w \min\{I(x_1,w), I(x_2,w)\}}{\sum_w \max\{I(x_1,w), I(x_2,w)\}} \ . \tag{8}$$

The similarity value obtained by this measure is higher as the number of highly informative features, providing comparable amount of information for both elements x_1 and x_2, is larger.

Lin, 1998 incorporates the information term of Equation 7, as well, though differently:

$$sim_L(x_1,x_2) = \frac{\sum_{\{w / I(x_1,w)>0 \wedge I(x_2,w)>0\}} (I(x_1,w) + I(x_2,w))}{\sum_w (I(x_1,w) + I(x_2,w))} \ . \tag{9}$$

Here, the obtained similarity value is higher as the number of features that are somewhat informative for both elements, x_1 and x_2, is larger, and the relative contribution of those is in proportion to the total information they convey.

Similarly to the cosine measure, both sim_{DMM} and sim_L measures satisfy: (i) the maximal similarity value, 1, is obtained for element pairs for which every feature is equally informative (including self similarity); and (ii) the minimal similarity value, 0, is obtained whenever every attribute is not informative for either one of the elements. Accordingly, our formulation and experiments below follow the convention that a zero value denotes no similarity (see Subsection 3.3 and Section 4).

In the coupled clustering experiments on textual data that are described later, we use both above similarity measures. We utilize pre-calculated sim_L values for one experiment (Subsection 5.1) and we calculate sim_{DMM} values, based on word co-occurrence within our corpora, for another experiment (Subsection 5.2).

[3] The expectation of the term given by Equation 7 over co-occurrences of all x's and w's, provided the unaltered $\log_2$ function is in use, defines the *mutual information* of the parameters x and w (Cover and Thomas, 1991).

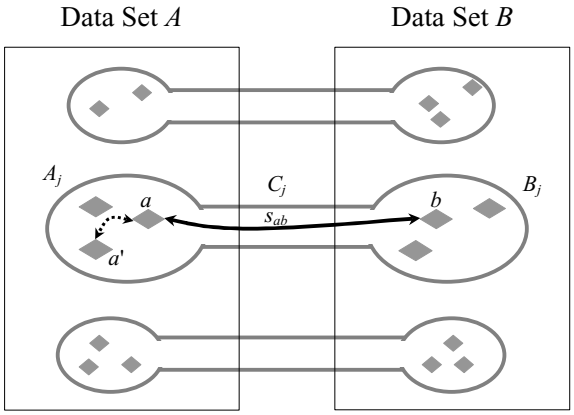

Figure 3: The coupled clustering setting. The diamonds represent elements of the given data sets A and B. The long arrow represents one of the values in use: a similarity value pertaining to two elements, one from each data set. The short arrow stands for one of the disregarded similarity values within a data set.

3. Algorithmic Framework for Coupled Clustering

In this section, we define the coupled clustering task and introduce an appropriate setting for accomplishing it. We then present alternative cost criteria that can be applied within this setting and describe the search method that we use to identify coupled-clustering configurations of low cost.

3.1. The Coupled Clustering Problem

As we note in Section 1, coupled clustering is the problem of partitioning two data sets into corresponding subsets, so that every subset is matched with a counterpart in the other data set. Each pair of matched subsets forms jointly a coupled cluster. A coupled cluster consists of elements that are similar to one another and distinct from elements in other clusters, subject to the context imposed by aligning the clustered data sets with respect to each other.

3.2. Pairwise Setting Based on Between-data-set Similarities

Coupled clustering divides two given data sets denoted by A and B into disjoint subsets $A_1, \ldots, A_k$ and $B_1, \ldots, B_k$. Each of these subsets is coupled with the corresponding subset of the other data set, that is A_j is coupled with B_j for $j = 1 \ldots k$. Every pair of coupled subsets forms a unified coupled cluster, $C_j = A_j \curlyvee B_j$, containing elements of both data sets (see Figure 3). We approach the coupled clustering problem through a pairwise-similarity-based setting, incorporating the elements of both A and B. Our treatment is independent of the method through which similarity values are compiled: feature-based calculations such as those described in Subsection 2.2, subjective assessments, or any other method.

The notable feature distinguishing our method from standard pairwise clustering, is the set of similarity values, S, that are considered. A standard pairwise clustering procedure potentially considers all available similarity values referring to any pair of elements within the single clustered data set, with the exception of the typically excluded self-similarities. In the coupled clustering setting, there are two different types of available similarity values. Values of one type denote similarities between elements within the same data set (*within-data-set similarities*; short arrow in Figure

3). Values of the second type denote similarities of element pairs consisting of one element from each data set (*between-data-set similarities*; long arrow in Figure 3). As an initial strategy, to be complied with throughout this paper, we choose to ignore similarities of the first type altogether and to concentrate solely on between-data-set similarities: $S = \{s_{ab}\}$, where $a \in A$ and $b \in B$. Consequently, the assignment of a given data element into a coupled cluster is directly influenced by the most similar elements of the other data set, regardless of its similarity to members of its own data set.

The policy of excluding within-data-set similarities captures, according to our conception, the unique context posed by aligning two data sets representing distinct domains with respect to one another. Correspondences, underlying presumed parallel or analogous structure of the compared systems, that are special to the current comparison are thus likely to be identified, abstracted from the distinctive information characterizing each system individually. Whether and how to incorporate the available information regarding within-data-set similarities, while maintaining the contextual orientation of our method is left to a follow up research.

3.3. Three Alternative Cost Functions

Given the setting described above, in order to identify configurations that accomplish the coupled clustering task, our next step is defining a cost function. In formulating it, we closely follow the standard pairwise-clustering framework presented by Puzicha, Hofmann and Buhmann, (2000, see Subsection 2.1 above). Given a collection of similarity values S pertaining to the members of two data sets, A and B, we formulate an additive cost function, $H(S,M)$, which assigns a cost value to any coupled-clustering configuration M. Equipped with such a cost function and a search strategy (see Subsection 3.4 below), our procedure would be able to output a coupled clustering configuration specifying assignments of the elements into a pre-determined number, k, of coupled clusters. We concentrate on Puzicha et al.'s H^0 cost function (Subsection 2.1, Equation 6), which is limited to similarity values within each cluster and weights each cluster's contribution proportionally to its size. Below we present and analyze three alternative cost-functions derived from H^0.

As in clustering in general, the coupled clustering cost function should assign similar elements into the same cluster and dissimilar elements into distinct clusters (as articulated by the monotonicity axiom in Subsection 2.1). A coupled-clustering cost function is thus expected to assign low cost to configurations in which the similarity values, s_{ab}, of elements a and b of coupled subsets, A_j and B_j, are high on average. (The dual requirement to assign low cost whenever similarity values of elements a and b of non-coupled subsets A_j and $B_{j'}$, $j \neq j'$, are low, is implicitly fulfilled). In addition, we seek to avoid influence of transient or minute components—those that could have been evolved from casual noise or during the optimization process—and maintain the influence of stable larger components. Consequently, the contribution of large coupled clusters to the cost is greater than the contribution of small ones with the same average similarity. This direction is realized in H^0 through weighting each cluster's contribution by its size.

In the coupled-clustering case, one apparent option is to straightforwardly apply the original H^0 cost function to our restricted collection of similarity values. The average similarity of each cluster is then calculated as

$$Avg'_j = \frac{\sum_{a \in A_j, b \in B_j} s_{ab}}{n_j \times (n_j - 1)} ,$$

where n_j is the total size of the coupled cluster C_j. (This is equivalent to setting to 0 all within-data-set similarities in Equation 5). As in H^0, the average similarity of each cluster is multiplied by the coupled-cluster size. Thus, the following cost function, H^1, is obtained:

$$H^1 = -\sum_j n_j \times Avg'_j \,. \tag{10}$$

Alternatively, since the calculations are limited to a restricted collection of similarities, we can incorporate the actual size of the similarity collection in use while averaging. The actual number of considered similarities in the restricted collection is, for each j, the product $n_j^A \times n_j^B$ of the sizes of the two subsets A_j and B_j composing C_j. The obtained averaging formula might seem more natural for the purpose of coupled clustering:

$$Avg''_j = \frac{\sum_{a\in A_j, b\in B_j} s_{ab}}{n_j^A \times n_j^B} \,,$$

Correspondingly, a second cost variant, H^2, is given:

$$H^2 = -\sum_j n_j \times Avg''_j \,. \tag{11}$$

However, the weighting scheme used within H^1 and H^2 treats each coupled cluster as a unified object. There might be some significance to the proportion of the subset sizes that are coupled within each cluster. Hence, we suggest yet another alternative: to weight the average similarity each cluster contributes to the cost by the geometrical mean of the corresponding coupled subset sizes: $\sqrt{n_j^A \times n_j^B}$. This yields our last cost function:

$$H^3 = -\sum_j \sqrt{n_j^A \times n_j^B} \times Avg''_j \,. \tag{12}$$

The weighting factor of H^3 results in penalizing large gaps between the two sizes, n_j^A and n_j^B, and in preferring balanced configurations, whose coupled-cluster inner proportions maintain the global proportion of the clustered data sets ($n_j^A / n_j^B \cong n^A / n^B$ for each j).

Puzicha, Hofmann and Buhmann, (2000) based their characterization of pairwise-clustering cost-functions on some properties and axioms (see Subsection 2.1 above). We have followed their conclusions in adapting, in three different variants, one function, H^0, that realizes the most favorable properties. It is worthwhile to see if and how these properties are preserved through the adaptation for the coupled clustering setting. All three cost functions obtained, H^1, H^2 and H^3, are additive (Equation 1) by construction. They also straightforwardly satisfy the scale invariance property (Equation 2). As for shift-invariance (Equation 3), except by H^2, this property is not fulfilled. However, the effect of a constant added to all between-data-set similarity values is bounded for H^1 and H^3, as well[4]. Finally, robustness (Equation 4) is satisfied by H^1 and H^3 (but not by H^2, see Appendix A).

Using the coupled sizes' geometrical mean as a weighting factor, H^3 tends to escape configurations whose clusters match minute subsets with large ones, which are

[4]To check shift-invariance, one can use the derivative of, say, H^3 with respect to Δ, which is the increment for all between-data-set similarity values. This is a linear function so the resulting derivative is $D = \square_j \sqrt{\sqrt{n_j^A \cdot n_j^B}}$. Consequently, normalizing H^3 by $1/D$ would result in perfect shift invariance. However, this function, in its non-normalized form, is (near) shift-invariance with regard to configurations for which the clusters (nearly) maintain the global proportion of the clustered data sets A and B, while highly imbalanced configurations are highly penalized. Since our experiments use similarity measures with values between 0 and 1, we stick to the simple formulation presented above, assuming that the normalized form would behave similarly.

occasionally the consequence of noise in the input data or of fluctuations in the search process. It turns that this property provides H^3 with a notable advantage over H^1 and H^2, as our experiments indeed show (see Sections 4, 5.2 and 6).

3.4. Optimization Method

In order to find the clustering configuration of minimal cost, we have implemented a stochastic search procedure, namely the Gibbs sampler algorithm (Geman and Geman, 1984). Starting with random assignments into clusters, this algorithm iterates repeatedly through all data elements and probabilistically reassigns each one of them in its turn, according to a probability governed by the expected cost change. Suppose that in a given assignment configuration, M, the cost difference $\Delta_{j/a,M}$ is obtained by reassigning a given element, a, into the j-th cluster ($\Delta_{j/a,M} = 0$ in case a is already assigned to the j-th cluster). The target cluster, into which the reassignment is actually performed, is selected among all candidates with probability

$$p(j) \equiv p(j|a,M) \propto \frac{1}{1 + \exp\{-\beta \Delta_{j|a,M}\}} .$$

Consequently, the chances of an assignment to take place are higher as the resulting reduction in cost is larger. In distinction from the similar Metropolis algorithm (Metropolis et al., 1953), assignments that result in increased cost are possible, though with relatively low probability. The β parameter, controlling the randomness level of reassignments, functions as an inverse "computational temperature". Starting at high temperature followed by progressive cooling schedule, that is initializing β to a small positive value and gradually increasing it (e.g., repeatedly multiply β by a constant that is slightly greater than one), turns most profitable assignments increasingly probable. As the clustering process proceeds, gradual cooling systematically reduces the probability that the algorithm would be trapped in a local minimum (though global minimum is fully guaranteed only under an impracticably slow cooling schedule). In our experiments, we have typically initialized β to the mean of the data set sizes divided by the cost of the initial configuration and multiplied β by a factor of 1.001, following every iteration in which improvement in cost has been achieved. The algorithm stops after several repeated iterations through all data elements, in which no cost change has been recorded (50 iterations in our experiments).

4. Experiments with Synthetic Data

A set of experiments on synthetic data has been conducted for comparing the performance of our algorithm, making use of the three cost functions introduced in Subsection 3.3 above. These experiments have measured, under changing noise levels, how well each of the functions reconstructs a configuration of pre-determined clusters of various inner proportions.

Each input similarity value (i.e. between-data-set similarities, see Subsection 2.2) in these experiments incorporates a basic similarity level, dictated by the pre-determined clustering configuration, combined with an added random component introducing noise. The basic similarity values have been generated so that each element is assigned into one of four coupled clusters. Elements in the same cluster share the maximal basic similarity of value 1, while elements in distinct clusters share the minimal basic similarity 0. The noisy component combined with the basic value is a random number between 0 and 1.

In precise terms, the similarity value s_{ab}, of any $a \in A$ and $b \in B$ (A and B are the clustered data sets), has been set to

$$s_{ab} = (1-x)\delta_{j(a)j(b)} + xr_{ab},$$

where $\delta_{j(a)j(b)}$—the basic similarity level—is 1 if $a \in A$ and $b \in B$ are, by construction, in the same (j-th) coupled cluster or otherwise 0 and r_{ab}—the random component—is sampled uniformly between 0 and 1, differently for each a and b in each experiment. The randomness proportion parameter x (i.e. level of added noise), also between 0 to 1, is fixed throughout each experiment, to keep the noise level of each experiment unaltered.

In order to study the effect of the coupled-cluster inner proportion, we have run four sets of experiments. Given data sets A and B consisting of 32 elements each, four types of synthetic coupled-clustering configurations have been constructed, in which the sizes n_j^A and n_j^B of the coupled subset pairs $A_j \subset A$ and $B_j \subset B$, together forming the j-th coupled-cluster, have been set as follows: (i) $n_j^A = n_j^B = 8$, for $j = 1\ldots4$; (ii) $n_j^A = 10$, $n_j^B = 6$ for $j = 1,2$ and $n_j^A = 6$, $n_j^B = 10$ for $j = 3,4$; (iii) $n_j^A = 12$, $n_j^B = 4$ for $j = 1,2$ and $n_j^A = 4$, $n_j^B = 12$ for $j = 3,4$; (iv) $n_j^A = 14$, $n_j^B = 2$ for $j = 1,2$ and $n_j^A = 2$, $n_j^B = 14$ for $j = 3,4$. These four configuration types, respectively labeled '8-8', '10-6', '12-4' and '14-2', have been used in the four experiment sets.

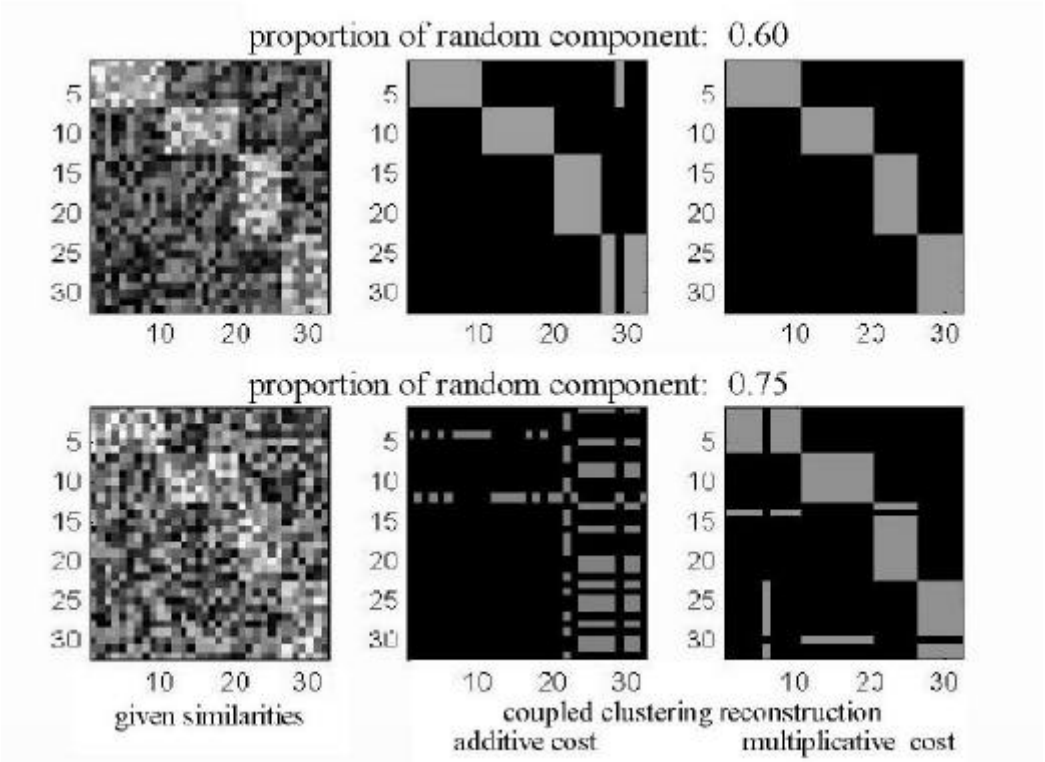

Figure 4: Reconstruction of synthetic coupled-clustering configurations of inner proportion '10-6' from noisy similarity data. Lines and columns of the plotted gray-level matrices correspond to members of the two sets. On the left-hand side—original similarity values—the gray-level of each pixel represents the corresponding similarity value between 0 (black) and 1 (white). In the reconstructed data, gray level corresponds to average similarity within each reconstructed cluster. The bottom part demonstrates that the multiplicative cost function, H^3, reconstructs better under intensified noise.

It is convenient to visualize a collection of similarity values as a gray-level matrix, where rows and columns correspond to individual elements of the two clustered data sets and each pixel represents the similarity level of the corresponding elements. The diagrams on the left-hand side in Figure 4 show two collections of similarity values generated with two different noise levels. White pixels represent the maximal similarity level in use, 1; black pixels represent the minimal similarity level, 0; the intermediate gray levels represent similarities in between. The middle and right-hand-side columns of Figure 4 display clustering configurations as reconstructed by our algorithm using the additive H^2 and multiplicative H^3 cost functions respectively, given the input similarity values displayed on the left-hand side. Examples from the 10-6 experiment set, with two levels of noise, are displayed. Bright pixels indicate that the corresponding elements are in the same reconstructed cluster. It demonstrates that, for the 10-6 inner proportion, the multiplicative variant H^3 tends to tolerate noise better than the additive variant H^2 and that this advantage grows when the noise level intensifies (bottom of Figure 4).

The performance over all experiments in each set has bean measured through *accuracy*, which is the proportion of data elements assigned into the appropriate coupled cluster. Since in cases of poor reconstruction it is not obvious how each reconstructed cluster associates with an original one, the best result obtained by permuting the reconstructed clusters over original clusters has been considered. Figure 5 displays average accuracy for the changing noise levels, separately for each experiment set. The multiplicative cost function H^3, is biased toward balanced coupled clusters, i.e. clusters in which the inner proportion is close to the global proportion of the data sets (which is perfectly balanced, 32-32, in our case). Our

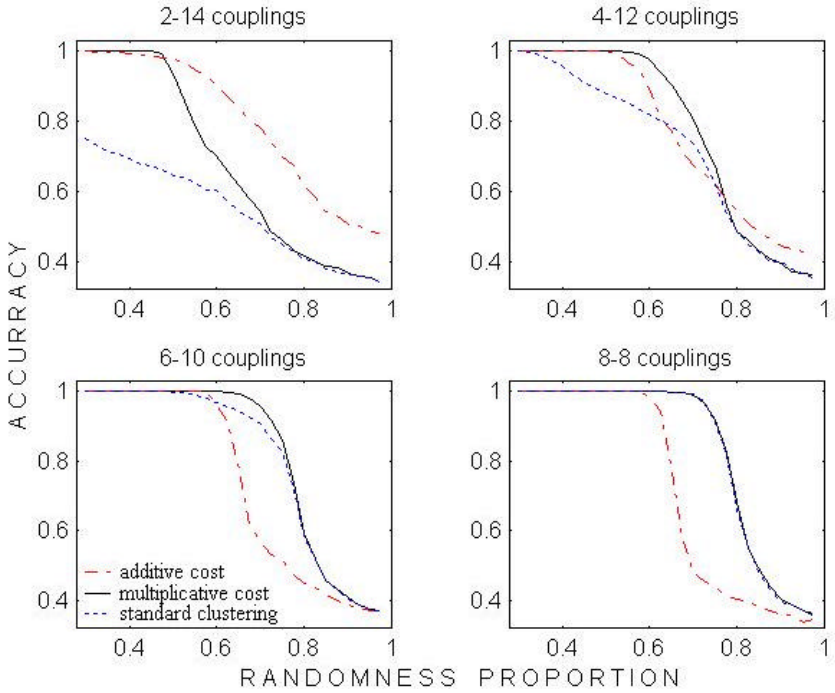

Figure 5: Accuracy as a function of the noise level (randomness proportion) for different coupled size proportions, obtained through experiments in reconstructing synthetic coupled-clustering configurations. For each proportion, results obtained using the restricted standard clustering (H^1), additive (H^2) and multiplicative (H^3) cost functions are compared.

experiments indeed verify that H^3 reconstruct better than the other functions, particularly in cases of almost balanced inner proportions.

Figure 5 shows that the accuracy obtained using the restricted standard-clustering function H^1 is consistently worse than the accuracy of H^3. In addition, for all internal proportions, there is some range, on the left-hand side of each curve, in which H^3 performs better than the additive function H^2. The range where H^3 is superior to H^2 is almost unnoticeable for the sharply imbalanced internal proportion (2-14) but becomes prominent as the internal proportion approaches balance. Consequently, it makes sense to use the additive function H^2 only if both: (i) there is a good reason to assume that the data contains mostly imbalanced coupled clusters and (ii) there is a reason to assume high level of noise. Real world data might be noisy, but given no explicit indication that the emerging configurations are inherently imbalanced, the multiplicative function H^3 is preferable. Consequently, we have used H^3 in our experiments with textual data, described in the following sections.

5. Coupled Clustering for Textual Data

In this section, we demonstrate capabilities of the coupled clustering algorithm with respect to real-world textual data, namely unstructured sets of keywords (corresponding to data sets *A*, *B* of Subsection 3.3). The keywords have been extracted from given corpora focused on distinct domains. Our experiments have been motivated by the target of identifying concepts that play similar or analogous roles in the examined domains. These experiments examine how well coupled clusters produced by our method reflect such conceptual equivalencies. Our results demonstrate meaningful and interesting semantic correspondences of themes, entities and categories revealed through coupled clustering.

Our setting assumes that the data sets are given or can be extracted automatically. We have used the TextAnalyst 2.0 software by MicroSystems Ltd.[5] to generate data sets for our experiments. This software can identify key-phrases in the given corpora. We have excluded the items that have appeared in fewer than three documents. Thus, relatively rare terms and phrases that the software has inappropriately segmented have been filtered out.

After extracting the data sets, between-data-set similarities, if not given in advance, should be calculated. In general terms, every extracted keyword is represented by a co-occurrence vector, whose entries essentially correspond to all co-occurring words (concrete examples follow in the subsections bellow), less a limited list of function words. Then, between-data-set similarity values are calculated using methods, such as those described in Subsection 2.2, to adapt the data for the coupled-clustering algorithmic setting introduced in Subsection 3.2. We differentiate between two optional sources that can provide the co-occurrence data for the similarity calculations. One option is to base the calculations on co-occurrences within the same corpora from which the keyword sets have been extracted. Thus, the calculated similarity values naturally reflect the context in which the comparison is being made. However, sometimes the compared corpora might be of small size and there is a need to rely on a more informative statistical source. An alternative option is to utilize the co-occurrences within an additional independent corpus for the required similarity calculations. In order to produce reliable and accurate similarity values, such independent corpus can be chosen to be significantly larger than the compared ones,

[5] An evaluation copy of TextAnalyst 2.0 is available for download at **http://www.megaputer.com/ php/eval.php3**.

but it is important that it addresses well the topics that are being compared, so the context reflected by the similarities is still relevant.

The following subsections provide results that have been obtained using the two approaches described above. In Subsection 5.1, the keyword sets come from news articles referring to two conflicts of different character that are nowadays in the focus of public attention. In this case, we make use of pre-given word similarity values. In Subsection 5.2, we turn to larger corpora focused on various religions. There, keywords and co-occurrence counts underlying similarity calculations are extracted from the same corpora.

5.1. Coupled Clustering of Conflict Keywords Using Pre-given Similarities

The conflict corpora are composed of about 30 news articles each (200–500 word tokens in every article), regarding the two conflicts mentioned in Section 1—the Middle East conflict and the dispute over music copyright—downloaded in October 2000.

We have obtained the similarities from a large body of word similarity values that have been calculated by Dekang Lin, independently of our project (Lin, 1998). Lin has applied the sim_L similarity measure (Subsection 2.2, Equation 9) to word co-occurrence statistics within syntactic relations, extracted from a very large news-article corpus.[6] We assume that this corpus includes sufficient representation of the conflict keyword sets in relevant contexts. That is: even if the articles in the corpus do not explicitly discuss the concrete conflicts, it is likely that they address similar issues, which are rather typical as news topics. In particular, occurrences of the clustered keywords within this corpus are assumed to denote meanings resembling their sense within our small article collection that might not provide sufficiently rich statistics for extracting this information due to its limited size.

As Table 1 shows, the coupled-clusters that have been obtained by our algorithm fall, according to our classification, within three main categories: “Parties and Administration”, “Issues and Resources in Dispute” and “Activities and Procedure”. To improve readability, we have also added an individual title to each cluster.

The keywords labeled “poorly-clustered”, at the bottom of Table 1, are assigned to a cluster with average similarity considerably lower than the other clusters, or for which no relevant between-data-set similarities are found in Lin's similarity database. Consequently, these keywords could be straightforwardly filtered out. However, poorly clustered elements persistently occur in most of our experiments and we include them here for the sake of conveying the whole picture.

Making use of pre-given similarity data is, on the one hand, trivially advantageous. Apart from saving programming and computing resources, such similarity data typically relies on rich statistics and its quality is independently verified. Moreover: in principle, pre-given similarity data could be utilized for further experiments in clustering additional data sets that are adequately represented in the similarity database. However, there are several disadvantages in taking this route. First, reliable relevant similarity data is not always available. In addition, the context of comparing two particular domains might not be fully articulated within generic similarity data that has been extracted in a much broader context. For example, the interesting case where the same keyword appears in both clustered sets, but it is used for different meanings, could not be traced. A keyword used differently in distinct corpora would co-occur with different features in each corpus . In contrast, when similarities are

[6] This corpus contains 64 million word tokens from Wall Street Journal, San Jose Mercury, and AP Newswire. The similarity data is available at **http://armena.cs.ualberta.ca/lindek/downloads/sims.lsp.gz**.

Table 1: Coupled clustering of conflict related keywords. Every row in the table contains the keywords of one coupled cluster. Cluster titles and titles of the three groups of clusters were added by the authors.

	Middle-East	Music Copyright
Parties and Administration		
Establishments	**city state**	**company court industry university**
Negotiation	**delegation minister**	**committee panel**
Individuals	**partner refugee soldier terrorist**	**student**
Professionals	**diplomat leader**	**artist judge lawyer**
Issues and Resources in Dispute		
Locations	**home house street**	**block site**
Protection	**housing security**	**copyright service**
Activity and Procedure		
Resolution	**defeat election mandate meeting**	**decision**
Activities1	**assistance settlement**	**innovation program swap**
Activities2	**disarm extradite extradition face**	**use**
Confrontation	**attack**	**digital infringement label shut violation**
Communication	**declare meet**	**listen violate**
Poorly-clustered keywords		
low similarity values	**interview peace weapon**	**existing found infringe listening medium music song stream worldwide**
No similarity values	**armed diplomatic**	

computed from a unified corpus, self-similarity is generally equal to the highest possible value (1 in Lin's measure), which is typically much higher than other similarity values. In such case, the two distinct instances of a keyword presenting in both clustered sets would always fall within the same coupled-cluster

5.2. Clustering of Religion Keywords with Dedicated Similarity Calculations

The other alternative for calculating similarity values is to use co-occurrence statistics from corpora that are focused on the compared domains, from which the clustered keywords can be extracted, as well. In this case, it is clear that each keyword appears in its relevant sense or senses. Hence, context dependent subtleties, such as identical keywords denoting different meanings, can be resolved. In this case, we rely on the assumption that there is a substantial overlap between the features, namely words commonly co-occurring in the two corpora, and that at least some of the overlapping features are used similarly within both. Specifically, we assume that the corpora to which we refer below—introductory web pages and encyclopedic entries concerning religions—contain enough common vocabulary directed towards some "average-level" reader, thus enabling co-occurrence-based similarity calculations that are fairly

informative. In summary, while the use of pre-given similarity data takes advantage of richer statistics over a unified set of features, the other alternative analyzes keywords in their more appropriate and accurate sense. It makes sense to use this approach whenever a sufficient amount of shared features and rich statistics are present, as exemplified below.

We have applied our method to corpora that discuss distinct religions in order to compare these religions to one another and to identify corresponding concepts within them. The religion data consists of five corpora, each of which focuses on one of the following religions: Buddhism, Christianity, Hinduism, Islam and Judaism. All documents, namely encyclopedic entries, electronic periodicals and additional introductory web pages, have been downloaded from the Internet. Each corpus contains 1–1.5 million word tokens (5–10 Megabyte).

Keywords that have been provided by comparative religion experts are included in the data sets, in addition to the keywords extracted by the TextAnalyst 2.0 software (the expert data has been primarily used for quantitative evaluation, see Section 6). The total size of each of the final keyword sets is 150–200, of which 15–20% were not extracted by TextAnalyst, but solely by the experts.

Each keyword has been represented by its co-occurrence vector as extracted from its own corpus. In counting co-occurrences, we have used two-sided sliding window of ±5 words, truncated by sentence ends (similarly to Smadja, 1993). On one hand, this window size captures most syntactic relations (Martin, Al and van Sterkenburg, 1983). On the other hand, this scope is wide enough to score terms that refer to the same topic in general—and not only literally interchangeable terms—as similar (Gorodetsky, 2001), which is in accordance with our aim of identifying corresponding topics. We have applied to the obtained vectorial representations the sim_{DMM} similarity measure, which incorporates detailed information on the data (Dagan, Marcus and Markovitch 1995; Subsection 2.2, Equation 8: sim_{DMM} incorporates maximal and minimal information values for each common feature individually; sim_L is less detailed in that it separately sums maximal values versus minimal values). After calculating between-data-set similarities, we ran the coupled clustering algorithm on each dataset pair.

In Table 2, we present the full coupled-clustering results for Buddhism versus Christianity. The keyword sets are partitioned into 16 coupled clusters, ordered by their average similarity in descending order. The poorly clustered elements, i.e. those contained in the 16th cluster with the lowest average similarity, are not shown. We attach intuitive titles to each cluster for readability and orientation. The obtained clusters (and also additional results that are not shown concerning other religion pairs) appear to reflect consistently several themes:

- Holy books, writing, teaching, and studying.
- Individual figures and their characteristics.
- Names of places and institutions.
- Ethics, emphasizing sins and their consequences.
- Traditions and schools.
- Festivals.
- Basic principles of each religion.

Table 2: Coupled clustering of Buddhism and Christianity keywords. Cluster labels were added by the authors. The 16th cluster of lowest average similarity is not shown.

	Buddhism	Christianity
1. Schools and Traditions	**doctrine, establish, ethic, exist, Hindu, India, Mahayana, scholar, school, society, study, Zen**	**catholic, history, Protestant, religion, tradition**
2. Scripture and Theology	**book, question, religion, text, tradition, west**	**apostle, bible, book, doctrine, Greek, Jew, john, question, scripture, theology, translate, write**
3. Sin / Suffering	**cause, death, Dukkha, pain**	**death, flesh, Satan, sin, soul, suffer**
4. Founder Characteristics	**being, Buddha, experience, meditation, monk, sense, teaching**	**believe, child, church, faith, find, god, Jesus Christ, Paul, pray, word**
5. Mental States	**animal, attain, awaken, awareness, Bodhisattva, consciousness, disciple, enlightenment, existence, karma, mindfulness, moral, nirvana, realm, rebirth, speech, Tantra, teach, word**	**baptism, experience, moral, problem, relationship, teaching**
6. Approach to the Religious Message	**find, hear, learn, problem**	**baptize, born, disciple, friend, gentile, hear, hell, judge, judgment, king, lost, love of god, Mary, preach, prophet, sacrifice, savior, sinner, story, teach**
7. Locations / Figures / Ritual	**country, king, monastery, Sangha, temple**	**angel, authority, city, Israel, Jerusalem, priest, saint, service, Sunday, worship**
8. Central Symbols and Values	**Dharma, god, peace, wisdom**	**bless, cross, earth, gift, heaven, holy ghost, kingdom, peace, resurrection, revelation, righteousness, salvation**
9. Studying	**ascetic, Bhikkhu, discipline, friend, Gautama, guide, philosopher, student, teacher**	**learn, minister, study, teacher**
10. Commands, Sin and Punishment	**anger, kill, law**	**adultery, command, commandment, forgiveness, law, punish, repentance**
11. Philosophical Concepts	**emptiness, foundation, four noble truths, phenomena, philosophy, soul, theory**	**argue, argument, foundation, humanity, incarnation, trinity**
12. Traditions and their Origins	**Asia, china, founded, Japan, north, nun, pilgrim, Theravada, Tibet**	**Baptist, bishop, establish, member, ministry, orthodox**
13. Holy Books	**discourse, history, Lama, mandala, Pali canon, Sanskrit, scripture, story, sutra, translate, Tripitaka, Vinaya, write**	**author, gospels, Hebrew, New Testament, Old Testament, passage, writing**
14. Customs and Rituals	**Amida, gift, mantra, purity, sacrifice, spirit, worship**	**atonement, confession, Eucharist, good works, pilgrim, reward, sacrament**
15. Figures, Festivals, Holy Places	**Dalai Lama, Korea, Sri Lanka, writing**	**Christmas, Easter, Good Friday, Isaiah, Luke, mass, Matthew, monk, Pentecost, Pope, Rome, university, Vatican**

Furthermore, the displayed keyword coupled clusters reveal differences and variations between the compared religions that seem to us of particular interest:

- **The different senses of the term 'law'.** The term 'law' is assigned into different clusters in the Christianity–Islam configuration (recall that a keyword appearing in more than one corpus is represented differently within the relevant

data sets; see Figure 6A). The particular assignments are related with different senses. In Christianity, the sense is of written law and holy books, such as the laws of religion. In Islam, it is related to philosophical laws that require learning and understanding. In the Buddhism–Islam comparison, the senses differ even more sharply (Figure 6B): in Buddhism the term law stands for the message of God or "law of nature". In Islam, it is related with social law, and with education, as well.

- **Family relations.** Family relations seem to be important in Islam. In the obtained Buddhism–Islam clustering configuration, the related cluster is associated with few additional Islamic terms, concerning inter-personal relationships and ethics. The Buddhist terminology seem to be less developed with respect to these aspects (Figure 6C).
- **Religion founder.** The key-figures of each religion are consistently assigned to a cluster associating them with prominent themes of the particular religion. Buddha, for instance, is characterized by 'being', 'meditation' and 'teaching', Jesus—by 'believing' and 'prayer' and Muhammad—by 'Quran' and 'worship'.

We provide further qualitative evaluation of results concerning comparison of religions in Section 6 below.

6. Evaluation

In order to evaluate textual coupled clustering results, such as those presented in Section 5 above, we have asked human subjects whose academic field of expertise is the comparative study of religions to manually perform a coupled clustering task. We have asked the experts to point, without any restriction, the most prominent equivalent aspects common to given pairs of religions. (To convey a broad notion of equivalency, we have included the following phrase in the instructions: "… features and aspects that are *similar*, or *resembling*, or *parallel*, or *equivalent*, or *analogous* in the two religions under examination…"). Then, every expert has been requested to specify some freely chosen representative terms that characteristically address the

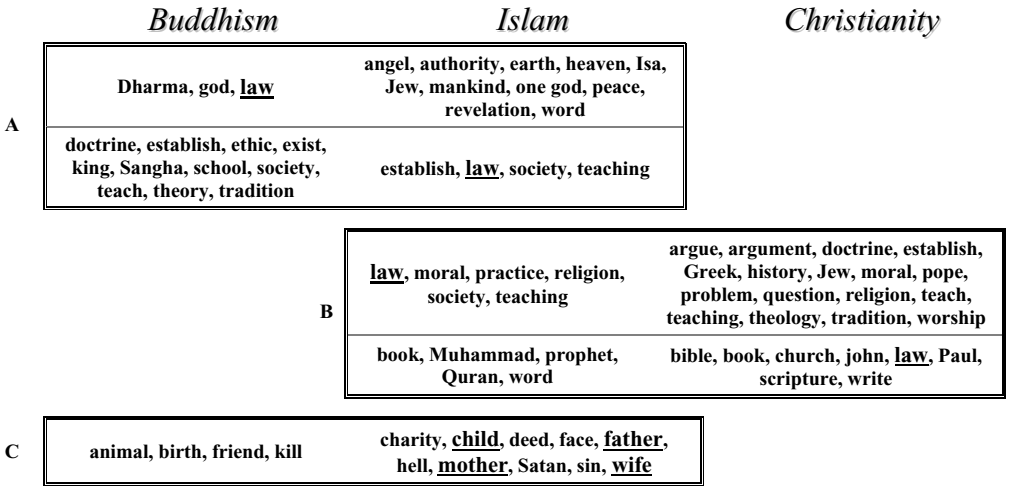

Figure 6: Excerpts from the Buddhism, Christianity, and Islam keyword coupled clustering, exemplifying different senses to the term 'law' (A, B) and the context that family-related terms obtain in the comparison of Islam and Buddhism (C).

identified similar aspects within the content world of each of the compared religions. For the purposes of the evaluation, the union of each resulting pair of corresponding sets of terms, separately addressing one aspect of similarity in two distinct religions, is defined to be a unified coupled-cluster. Example of aspects shared by Buddhism and Islam, as suggested by one of the experts, together with the associated terms specified by this expert can be seen in the first lines of Table 3.

Coupled clustering is a newly defined computational framework and no well-established means for evaluating its outcome are available. In the absence of objective criteria, we have found it appropriate to evaluate our procedure by measuring how well it approximates the keyword clustering-configurations produced by the human experts. We have let the participants choose the number of clusters for each religion pair, as well as the specific terms included in every cluster. Thus, our evaluation is free from bias that could have emerged due to restricting the experts to certain aspects of similarity or to a given list of keywords extracted in one method or another. For evaluation purposes, the two keyword sets, to be clustered by the computerized procedure in each case, have included the terms presented by the experts, excluding terms that are absent or rarely found in our corpora. These rare terms have been excluded also from the expert clusters, to allow unified grounds for comparison. The required similarity values have been calculated based on co-occurrence data extracted from the religion corpora (as in Subsection 5.2).

The measure of overlap between expert and automated clustering configurations is based on counting pairs of keywords consisting of one element from each of the clustered keyword sets. Specifically, we have used the Jaccard coefficient, which is in common use for evaluating information retrieval and clustering results (e.g., Ben-Dor, Shamir, & Yakhini, 1999). In the coupled clustering case, given clustering configurations by both expert and our computerized procedure, the Jaccard coefficient is defined as

$$\frac{n_{11}}{n_{11} + n_{10} + n_{01}},$$

where

n_{11} – the number of pairs of keywords, one of each data set, that have been assigned by both the expert and our program into the same cluster;

n_{01} – the number of keyword pairs that have been assigned into the same cluster by the expert but not by our program;

n_{10} – the number of keyword pairs that have been assigned into the same cluster by our program but not by the expert.

Combining in pairs the five religions mentioned in Subsection 5.2 above—Buddhism, Christianity, Hinduism, Islam and Judaism—make a total of ten religion pairs. We have obtained coupled clusters from three experts. One of the experts has contributed coupled-clustering configurations concerning all ten pairs, another one—three configurations, and the last one—four configurations. In total, we have 17 coupled-clustering configurations contributed by experts. Most clustered sets have consisted of 20–30 keywords.

The Jaccard coefficient is used to check the overlap of the clustering configurations obtained using the multiplicative cost function H^3 (Subsection 3.3, Equation 12; H^3 has been used also in Section 5) with expert configurations. This overlap level is compared to a simple benchmark: the expected coefficient obtained from the overlap between the experts configurations and configurations consisting of random assignments of elements into clusters. In addition, we compare the overlap of the expert configurations with results obtained using the additive cost function H^2 (Equation 11) to experts vs. multiplicative cost function overlap and experts vs. random assignment overlap. The overlap between different experts has also been recorded in all cases where different experts have provided configurations referring to the same pair of religions.

Figure 7 displays a sample of the Jaccard coefficient values measuring the overlap of six of the 17 configurations contributed by religion experts with the corresponding configurations due to the multiplicative and additive functions, as well as random assignments. The selection of six configurations has been chosen to represent, in a conveniently displayable manner, the various trends that were found with regard to the overlap of our automated configurations of 2 to 10 coupled clusters with the expert data. The displayed Jaccard coefficient values demonstrate that there is no a-

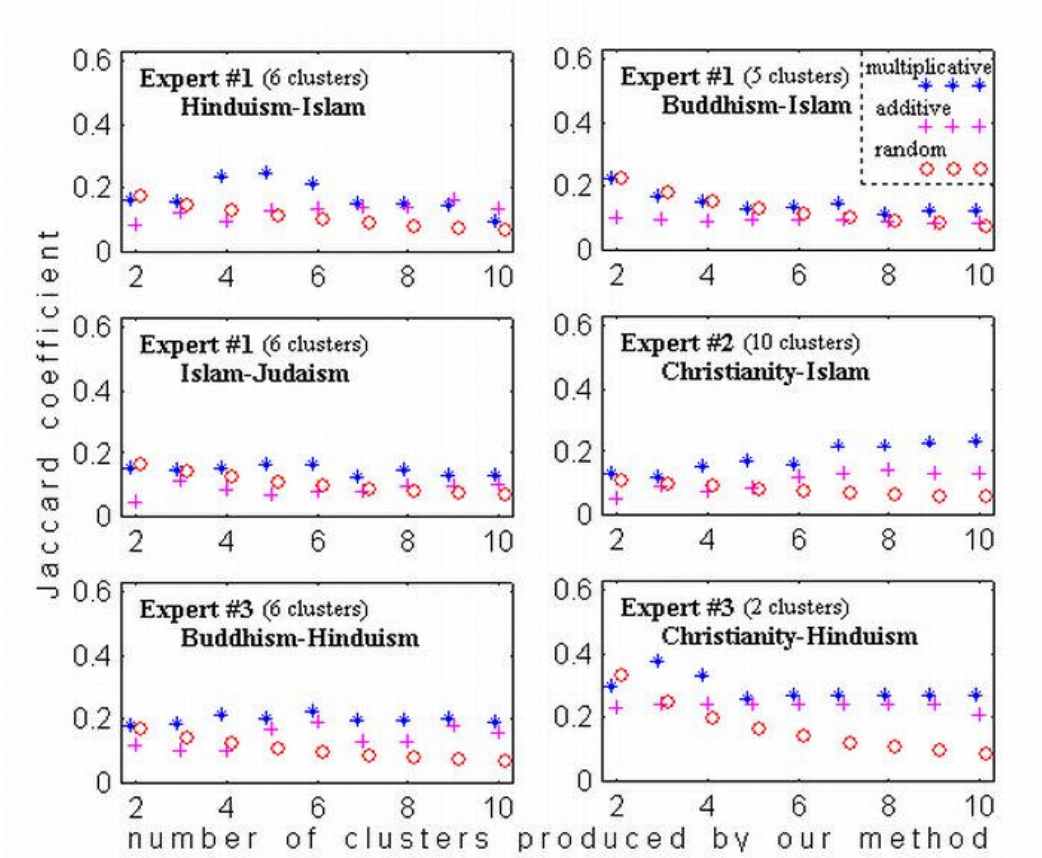

Figure 7: Jaccard coefficient values measuring overlap of religion expert configurations with configurations of 2–10 coupled clusters produced by our method—using the multiplicative, H^3, and additive, H^2, cost functions—and random assignments. A sample of results is shown, regarding six clustering configurations randomly sampled from the 17 configurations contributed by experts. The number of expert clusters is indicated in brackets.

priory known optimal number of clusters that would yield the best results. The optimum is obtained for various cluster numbers, which are not tightly related to the number of expert clusters (the cause of this empirical mismatch requires yet a further examination). However, over the whole displayed selection, we see that the multiplicative cost function in general shows superior performance compared to both additive function and random assignments. In some cases, there is a local maximum on the multiplicative function curve, additional to the global maximum, indicating that there is more than one meaningful resolution fitting the data.

In Figure 8, the Jaccard coefficient is averaged, separately for each number of our output clusters, over all comparisons to expert configurations (each with its own fixed number of expert clusters). Thus, we ignore optimal numbers of clusters obtained for each particular comparison (demonstrated for six cases by Figure 7). The overlap of expert configurations with random assignments decreases with the number of clusters, since randomness trivially dominates, as more clusters are added. In distinction, overlap with the additive cost configurations increases with the number of clusters. The additive function performs poorly in cases of fewer and larger clusters, because of its bias towards imbalanced couplings. The multiplicative function is shown to maintain its superiority consistently over the whole range. Figure 8 displays also some indication of variability among experts. Note that there have been only eight cases of identical religion pairs compared by distinct experts. In each of these cases, the overlap is measured on keyword sets that are considerably smaller than the original expert sets, since the distinct expert configurations do not share large proportion of common terms. It can be seen that one standard deviation below the average overlap among the experts is close to the average overlap level obtained by the multiplicative function.

For overall quantitative assessment of the results, we average over the Jaccard coefficient values obtained with 2 to 10 clusters, separately for each expert configuration and each assignment method. Thus, three corresponding sets are obtained, each related with another assignment method, containing 17 representative values each, which are assumed independent within each sample. We hypothesize that the difference between the corresponding values in these three sets is not coincidental. The difference between multiplicative cost and random average results, taken individually for each expert clustering configuration, is indeed statistically significant ($\bar{x} = 0.067$, $t(16) = 5.53$, $p < 0.00005$). Superiority of the multiplicative cost on the additive cost is even more significant ($\bar{x} = 0.070$, $t(16) = 13.98$, $p < 10^{-8}$), because these cost functions involve the same similarity data, they

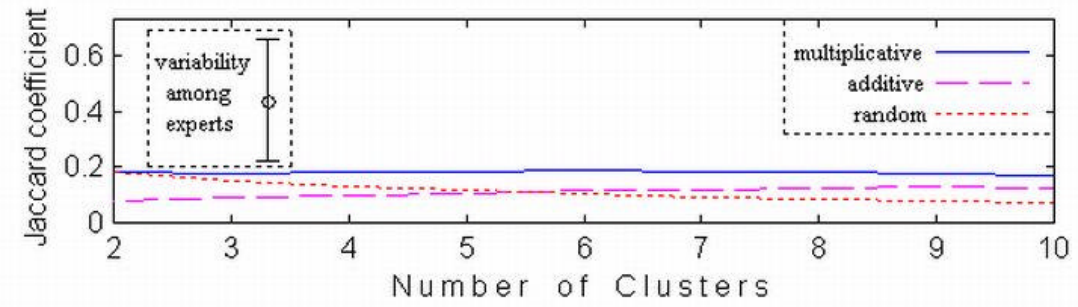

Figure 8: Average trends of three reconstruction methods—multiplicative H^3, additive H^2 and random—as a function of number of clusters produced by our method (independently of the number of expert clusters). Values are averaged over all 17 expert configurations, for every individual cluster number.

Table 3: A coupled clustering configuration regarding the mapping between Buddhism and Islam contributed by one of the experts (A) followed by the output of computerized procedures: coupled clustering using the multiplicative, H^3 (B) and additive, H^2 (C) cost functions and standard clustering, H^0 (D) – applied to the same keywords.

	Buddhism	Islam
A. Expert coupled clustering		
Scripture	**koan, mantra, mandala, pali canon, sutra**	**hadith, muhammad, quran, sharia, sunna**
Beliefs and ideas	**buddha nature, dharma, dukkha, emptiness, four noble truths, nirvana, reincarnation**	**allah, five pillars, heaven, hell, one god**
Ritual, prayer and festivals	**gift, meditation, sacrifice, statue, stupa**	**charity, fasting, friday, id al fitr, kaaba, mecca, pilgrim, pray, ramadan**
Mysticism	**samadhi, tantra**	**sufi**
B. Multiplicative cost function		
	dharma dukkha, meditation	**allah, muhammad, pray, quran**
	stupa	**kaaba**
	gift, nirvana, sacrifice	**charity, fasting, heaven, ramadan**
	emptiness, four noble truths, mandala, sutra, tantra	**hadith, mecca, pilgrim, sufi, sunna**
	pali canon, reincarnation, statue	**friday, hell, id al fitr, sharia**
	buddha nature, koan, mantra, samadhi	**five pillars, one god**
C. Additive cost function		
	dharma	**allah, mecca, muhammad, pray, quran**
	stupa	**kaaba**
	dukkha, emptiness, four noble truths, meditation, nirvana, pali canon, sutra, tantra	**hadith**
	reincarnation	**hell, one god**
	gift	**charity, fasting, friday, heaven, id al fitr, pilgrim, ramadan, sharia, sufi, sunna**
	buddha nature, koan, mandala, mantra, sacrifice, samadhi, statue	**five pillars**
D. Standard clustering (including within data set similarities)		
		fasting, Ramadan
		mecca, pilgrim
		heaven, hell
		allah, hadith, muhammad, pray, quran
	harma, dukkha, emptiness, four noble truths, meditation, nirvana, reincarnation	
	buddha nature, gift, koan, mandala, mantra, pali canon, sacrifice, samadhi, statue, stupa, sutra, tantra	**charity, five pillars, friday, id al fitr, kaaba, one god, sharia, sufi, sunna**

hence depend more heavily on each other. The difference between the corresponding random values and the values due to the additive cost function is not significant ($\bar{x}$ = 0.003, $t(16) = 0.26$, $p > 0.1$). (Figure 8, however, demonstrates that the last two assignment methods follow opposing trends once the number of clusters is taken into account).

Table 3 demonstrates the results described above. The top of the table displays a specific coupled clustering configuration, pertaining to Buddhism and Islam, contributed by one of the experts. It is followed by the output of computerized procedures applied to the same keywords. The next two configurations have been produced by our coupled clustering algorithm, making use of the multiplicative H^3 and additive H^2 cost functions respectively. Next, for further comparison, we have incorporated the usually disregarded within-data-set similarity values and applied to the union of the two keywords sets a standard clustering algorithm using the original cost function H^0 by Puzicha et al. (2000). Although the expert configuration consists of 4 clusters, the most convincingly interpretable results, which are displayed in Table 3, have been obtained with six clusters. The table shows that all attempts to reconstruct the expert configuration are imperfect. There are several of the expert clusters—in particular small ones, e.g., the one titled "mysticism"—for which no trace is found in the various computerized outputs. On the other hand, computerized configurations display some level of topical coherence, unrelated with the expert clusters, for example, the cluster that relates *nirvana* and *heaven* with *sacrifice* and *charity* in the multiplicative cost configuration. Standard clustering is demonstrated particularly ineffective in producing coupled-clustering configurations (and consequently it is not evaluated further). On the other hand, the additive cost function does provide several "hits", but it is biased towards imbalanced coupled-clusters, with accordance to our findings in this section and in Section 4 above. Finally, the multiplicative function produces seemingly qualitative outcome, also in this demonstrative example.

It should be noted that although our subjects have relied on their scholarship rather than on occasional knowledge, their expertise in religion studies does not eliminate the subjectivity inherent to the task they have performed. Comparative studies of religions do aim at comparing systematically distinct religions, but there are no theoretical grounds precisely specifying one certain coupling of equivalent themes. This type of complication might be even more prominent in evaluating coupled clustering in the general case, which might often deal with transitory comparisons. Indeed, the significance of inter-domain structural mapping to disciplines such as cognitive modeling and information sciences, is related to its association with mental processes such as discovery and creation (Gentner, 1983). Hence, coupled clustering might substantially lack proven recipes for the "right" solution, even more than other unsupervised methods such as standard clustering. In this section, we have found significant support to our results, with reference to data provided by experts that are, expectedly, not in close agreement with each other. Given the high subjectivity inherent to the keyword coupled clustering task, we find the results encouraging.

7. Related Work

Previous computational frameworks for detecting analogies and structural equivalencies have been motivated by cognitive considerations and related research and modeling. The algorithmic framework that implements the structure mapping theory (which has provided an initial motivation to our work, Gentner, 1983; see Section 1), is called the Structure Mapping Engine (SME, Falkenhainer, Forbus and

Gentner, 1989). It processes structured representative items digesting assumed knowledge regarding two distinct systems to be mapped to one another. A typical example for an item of the type SME processes is

GREATER-THAN [TEMPERATURE (coffee), TEMPERATURE (ice-cube)] ,

expressing the fact that the temperature of coffee is greater than that of an ice cube. Given such items referring to two systems, SME uses a sub-graph-match mechanism directed by the
requested similarity mode (feature-based or structural) to present an overall optimal match between the analyzed systems. SME has been criticized as bypassing issues that are in practice crucial to analogy making (Chalmers, French and Hofstadter, 1995, pp. 182–185). According to its criticizer's viewpoint, it does not incorporate interaction with lower level processes, perception for instance, which might considerably influence both the manipulated representations and the resulting structure mapping. Related to this criticism is our practical concern regarding availability of pertinent representations. Computational frameworks such as SME typically manipulate previously structured data. Hence, they require considerable pre-processing. In many cases, semantically structured representations are not readily available. In contrast, the input of our procedure, namely similarity values, can be produced given, e.g., co-occurrence records of any data.

A different style in detecting analogous structure is demonstrated by COPYCAT (Hofstadter and Mitchell, 1995), a computer program operating in the toy domain of letter strings. COPYCAT answers questions such as "if a source string *abc* is transformed into *abd*, what would be the analogously transformed value of a target string *xyz*". Here, the input data does not consist of ready-made representations, but it is processed according to domain-specific rules, e.g., matched successor relationships, matched increasing and decreasing sequences and so on. These rules are stochastically activated in order to construct cumulatively sensible associations between the source string to its transformed value and to the target string. The significance of COPYCAT is in simulating cognitive processes on a demonstrative level. It seems that formalizing the appropriate rules for an arbitrary domain might require utilization of pre-given knowledge and a specializing pre-processing stage. Coupled clustering, in comparison, suggests directions that do not depend on a specific domain.

Our perspective has been inspired by the above cognitive approaches, particularly in inspecting the relevance of internal details, which are retailored when an examined object is compared with different references. However, our framework thoroughly deviates from theirs. On one hand, our actualization of the notion of structure is relatively simplistic. We accomplish our target through aligning elementary clusters rather than graph-based constructs or domain-specific evolved representations. On the other hand, we assume simpler input as well: simple associations (similarities) between data elements, which can be efficiently applied to unrestricted types of real world data. Our method currently lacks the ability to deduce implicit target components, through unmatched patterns within the system for which richer information is present (such as COPYCAT's transformed sequence, given for the source sequence only). This direction can be followed in future elaboration of our framework.

Coupled clustering is a novel unsupervised computational task, distinct from the well-studied standard data clustering. There are, however, several standard clustering methods (i.e. clustering applied to a single data set) that combine clustering with additional aspects of structure. In a review of relational data-clustering methods used within social sciences, Batagelj & Ferligoj (2000) provide examples of combining clustering with relations that are embedded within the data. One of the reviewed methods—*blockmodeling*—seeks to cluster units that have substantially similar patterns of relationships with other units, and to interpret the pattern of relationships among clusters. There are several types of relational pattern similarity, e.g., *structural equivalence*, where elements are connected to the rest of individual elements in identical ways, and *regular equivalence*, where elements are equally or approximately connected to equivalent other elements. Another approach reviewed by Batagelj & Ferligoj (2000)—*constrained clustering*—groups similar units into clusters based on attribute data, but clusters have to satisfy also some additional conditions. For example: clusters of geographical regions that are similar according to their socioeconomic development level have to be determined such that the regions inside each cluster are also geographically connected. These approaches demonstrate how several criteria in clustering of a single data set can be combined. Their adaptation for coupled clustering framework could be helpful: for instance, the present associations (similarities) between data sets could still be utilized, while additional within-data-set criteria (such as syntactic relations within text) are added.

The policy adopted within coupled clustering, of restricting the processed data to associations (similarities) relating two distinct data sets, bears in mind previous works on *dyadic data* (Hofmann, Puzicha and Jordan, 1999). The dyadic setting virtually refers to data sampled or observed in pairs consisting of elements of two distinct classes. It applies, for instance, to verbs and their corresponding transitive subjects (Pereira, Tishby & Lee, 1993), documents and words they contain (Slonim and Tishby, 2000a and 2000b; Deerwester et al. 1990) authoritative and "hub" web pages (Kleinberg, 1999), and so on. Often, only one of the sampled classes forms the data elements to be processed (e.g., clustered) while elements of the other class are regarded as features. However, it has been demonstrated in several studies that data elements and features can switch roles. For example, Slonim and Tishby (2000a) cluster at first words with respect to the documents in which they appear and then use the word clusters as features to obtain improved document clustering. Similarly, principal component decomposition, which has been applied in many works (e.g., Deerwester et al. 1990; Kleinberg, 1999), results in low-dimensional representation of the two examined classes in terms of each other. This resembles our target of presenting entities of one domain—say, concepts of a given religion—in the terms of another domain. The most notable distinction between the dyadic setting and ours is that it refers to raw co-occurrence data, while our data consists of similarity values that have been obtained through pre-processing. In the experiments that we have actually conducted, calculating these values relies on a third set of features and the elements of any of the data sets cannot be interpreted as features of the other. In addition, the above-mentioned works typically depict soft relationships between the two examined classes (or otherwise alternated the roles of data and features providing no explicit reciprocal map, as in the case of Slonim and Tishby, 2000a). The outcome most similar to our configurations—a matrix of probabilistic co-occurrence relations of noun and adjective clusters—has been produced by the two-sided clustering model of Hofmann, Puzicha and Jordan, (1999). The setting of (hard) coupled clustering fits the goal of identifying analogous themes and producing reciprocal structure mapping.

Thus, our method outputs one-to-one subset correspondences rather than distributed relationships obtained from probabilistic approach. Extending our approach to distributional clustering seems plausible but it should be developed elsewhere.

8. Relational Commonalities

Incorporating details that are specific to the context of comparison takes coupled clustering beyond the ordinary surface-level feature-based type of similarity. However, in comparison to our simplistic notion of structure mapping, which is essentially partitioning of the data elements into matched subsets, cognitive approaches consider more sophisticated structural constructs incorporating compound relations that characterize each system. Structure mapping theory (Gentner, 1983) emphasizes the role, within high-level mental functioning, of *relational commonalities*, i.e. cross-system match of relations between entities. We sketch below a tentative procedure that we demonstrably use to extend the present coupled clustering framework towards detecting relational commonalities.

To start, assume that the coupled clustering procedure has already output a clustering configuration, M, of k coupled clusters. Assume further that the input similarities are based on co-occurrence counts, collected in the dyadic setting. In such case, we may expect that there is available information regarding the features commonly characterizing each coupled cluster. In the textual example given below, which utilizes a configuration obtained in our previous experiments (Section 5), we use counts of common words co-occurring in both data set keywords. The proposed procedure for detecting relational commonalities then follows the stages specified below:

1. Every occurrence of a data element is replaced by a label from '1', '2', …, 'k' designating its coupled cluster in M.
2. Existing co-occurrence data is used once more, but now data elements and features switch roles:
   - The new (single) data set consists now of former features previously used for calculating similarities. We have picked the most common ones to come up with a data set of reasonable size (1000 words). However, since most features occur frequently within either one of the corpora, each former feature can keep track of the corpus in which it occurs relatively frequently.
   - The cluster labels, replacing the original data elements, would now be regarded as features.
   - Similarities between the new data elements (i.e. former features) are being calculated, based on the new features (i.e. cluster labels).
3. Those elements co-occurring mainly with only one of the coupled cluster labels are filtered out from the obtained set of former features (they convey no information regarding relational commonalities).
4. The remaining former features are clustered by a cost-based standard clustering procedure (we have used the original H^0 cost function of Puzicha et al, 2000; Section 2).
5. Clusters that consist exclusively of former features that occur frequently in one of the corpora but not in the other are further filtered out from the resulting clustering configuration.

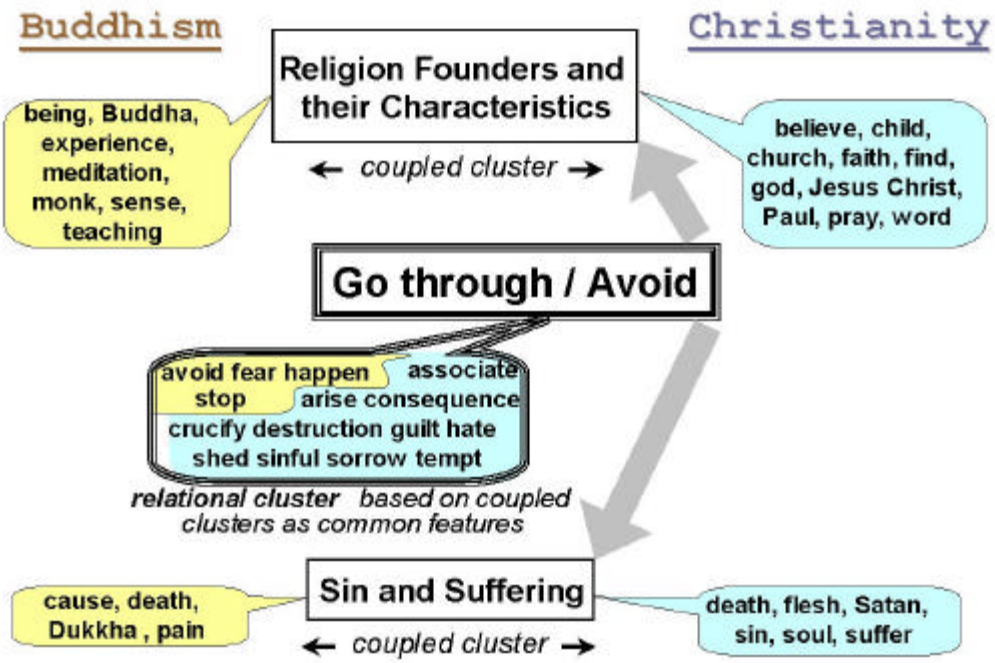

Figure 9: An illustration of the relational pattern—*RELIGION FOUNDER : GOES THROUGH : SUFFERING*—common to Buddhism and Christianity. The word-clusters that reveal this relation were extracted by a preliminary extension of the coupled clustering procedure (see text). The cluster labels were added by the authors. Within the relational cluster in the middle, different background color keeps track to the corpus in which each word appear more frequently.

The elements of the remaining clusters consistently co-occur, as former features, with more than one coupled cluster in both corpora. Hence, it is possible that they characterize some common relations among the clusters with which they co-occur. Preliminary experimentation with this procedure has identified some interesting feature clusters that underlie relations between coupled clusters from the configurations regarding religion comparisons. One such cluster that we have titled "Go through /Avoid" is illustrated in Figure 9. It associates the coupled clusters that we have titled "Religion Founders and their Characteristics" and "Sin and Suffering". Our interpretation of the relation between those coupled clusters reveals a pattern of a religion founder going through suffering (regardless of whether he admire the suffering or admits the need to avoid it). This manifestly demonstrates a common pattern of Buddhism and Christianity: both Jesus Christ and Buddha have extensively brought up the role of suffering, largely reflecting on the fundamentals of those religions. Furthermore, this relational commonality is unique to those two particular religions and it is not as prominent in Islam, for instance.

It has been mentioned in the previous section that features and data elements in the dyadic setting are interchangeable. Here we realize that this role switching carries added implication to the coupled clustering setting. The formed feature clusters provide unified reference to distinct systems that were first unrelated.

It should be emphasized that this procedure is preliminary and illustrative. Among the open issues that need to be investigated is how to filter out automatically the irrelevant clusters, which do not seem to convey information on common conceptual relations. However, the example above provides a detailed guideline to treatment of non-trivial structural aspects within our clustering-based approach.

9. Conclusions and Future Work

This paper presents coupled clustering, which is a newly defined unsupervised task that outputs configurations of matched subsets of two given data sets. We have motivated and demonstrated coupled clustering as a tool for revealing structural correspondence of distinct systems that are not necessarily similar at surface-level. The method has been successfully applied to synthetic and textual data. Significant overlap has been achieved with human experts of comparative religion studies that have performed manually the similar task within their field of expertise. Likewise, coupled clustering might be found relevant to cognitive scientists, information specialists, negotiators and historians, as well (see Section 1).

Classification and clustering are fundamental aspects of possibly latent cognitive processes. The coupled clustering method goes a step further, attempting to reveal both conformities and differences between distinct sources, such as an infant picking cues from her two parents while acquiring language proficiency. This last example appropriately motivates our approach but it also identifies several limitations of the current framework and suggests future directions for further study. The current setting is limited to two sources of information, but a child regularly utilizes the utterances produced by additional relatives, friends and TV programs for enriching lingual skills. Similarly, a desirable extension would be enabling our framework to process a greater number of distinct data sources. Furthermore, the current algorithm assumes given similarity values. It could be adapted to use directly co-occurrence data, similarly to the above-mentioned dyadic data methods (see Section 7). Another future direction is to make the coupled clustering framework interactive through added partial supervision. For example: computing the same cost function for configurations in which the user presets some of the between-data-set links.

Several issues that are relevant to data-clustering methods in general are applicable to coupled clustering as well. It has been shown (Gorodetsky, 2001) that clustering results can be enhanced through repeated iterations, while the input data is slightly modified, e.g., by perturbing the similarity values or by systematic exclusion of data elements (*covering design*, Nurmela, 1993). This can be particularly effective for coupled clustering, where the requested structure might be relatively unstable and there are no tested ways to validate the results. In data clustering, it is also desirable to identify outliers that do not properly belong in any cluster. In our current setting, we exclude the cluster of lowest average similarity, to obtain practically acceptable outcome (see Subsection 5.1). However, this point is of special interest to coupled clustering, where it is important to restrict the results to definite correspondences. Hence, further investigation of the confidence level attached to the assignment of each element into a coupled cluster is required. An additional aspect requiring additional investigation is how to determine the optimal number of clusters. A conceivable strategy to tackle this issue is picking a number of clusters with significantly lower cost in comparison to smaller numbers.

We have provided an initial demonstration of how our framework could be extended to tackle relational structure (Section 8). However, this route requires intensive further research. Finally, it would be interesting to use parameters of the obtained clustering, such as the final cost value, to devise an overall measure of structural correspondence and to compare it to conventional "surface-level" similarity measures.

To summarize, coupled clustering demonstrates the capabilities of unsupervised statistical methods in a setting assuming the presence of structural equivalence. Thus, it equips previous structure-based cognitive frameworks with a different viewpoint and a novel computational tool. Moreover, coupled clustering enriches the conventional machine learning methods by incorporating the effect of structure on depicting similarity between composite objects. Likewise, it demonstrates how comparing composite objects affects details within their structure. From either perspective—structural cognitive models or unsupervised learning—coupled clustering suggests rich grounds for further research and application.

Acknowledgments

We greatly appreciate the help and guidance that have been provided to us by Jan Puzicha. We thank William Shepard, Tiina Mahlamäki, Ilkka Pyysiainen, Rudi Siebert and Eitan Reich for responding to our survey and for illuminating discussions regarding the domain of comparative religion studies.

This work was partially supported by ISRAEL SCIENCE FOUNDATION founded by The Academy of Sciences and Humanities (grants 574/98-1 and 489/00). The work was also partially supported by the Germany Israel Foundation (GIF contract I 0403-001 06/95).

Appendix A: Formal Verification of Properties of the Cost Functions

The following proposition relates the cost functions H^1, H^2 and H^3, introduced in Subsection 3.3 (Equations 10, 11 and 12) with the robustness property (Subsection 2.1, Equation 4).

Proposition *H^1 and H^3 are robust; H^2 is not robust.*

We show the robustness of H^3. The other cases can be inspected similarly. Given a collection S of all between-data-set similarities concerning the elements of two sets A and B, $S^{a'+\Delta}$ denotes the collection resulted from adding Δ to all similarity values in S concerning one particular element, $a' \in A$. In addition, we denote by j' the index of the coupled cluster to which a' is assigned in the given configuration. Notice that there are $n^B_{j'}$ similarity values that are altered by the transformation from S to $S^{a'+\Delta}$.

$$\frac{1}{n^A+n^B}\left|H^3(M,S)-H^3(M,S^{a'+\Delta})\right| =$$

$$\frac{\sqrt{n^A_{j'}\times n^B_{j'}}}{(n^A+n^B)(n^A_{j'}\times n^B_{j'})}\left(-\sum_{b\in B_{j'}} s_{a'b}+\sum_{b\in B_{j'}}(s_{a'b}+\Delta)\right) =$$

$$\Delta\frac{\sqrt{n^B_{j'}}}{(n^A+n^B)\sqrt{n^A_{j'}}} \le \frac{\sqrt{n^B}\Delta}{n^A+n^B},$$

which vanishes whenever either n^A or n^B or both tend to infinity.

References

V. Batagelj and A. Ferligoj. Clustering relational data. In W. Gaul, O. Opitz, M. Schader, editors. *Data Analysis*: *Scientific Modeling and Practical Application*, pages 3–15, Springer, Berlin, Germany, 2000.

A. Ben-Dor, R Shamir and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology* 6(3/4):281–297, 1999.

D. Chalmers, R. French and D. Hofstadter. High-level perception, representation and analogy: A critique of artificial intelligence. In D. Hofstadter and the Fluid Analogies Research Group, *Fluid Concepts and Creative Analogies*, pages 169–193, Basic Books, New York, New York, 1995.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, New York, 1991.

I. Dagan, S. Marcus and S. Markovitch. Contextual word similarity and estimation from sparse data. *Computer Speech and Language***,** 9(2):123–152**,** 1995.

G. Das, H. Mannila and P. Ronkainen. Similarity of attributes by external probes. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining KDD'98*, pages 23–29, AAAI Press, New York, New York, 1998.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

B. Falkenhainer, K. Forbus and D. Gentner. The structure mapping engine: Algorithm and example. *Artificial Intelligence*, 41(1):1–63, 1989.

W. Gale, K. Church and D. Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439, 1993.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

D. Gentner. Structure-Mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

M. Gorodetsky. *Methods for Revealing Semantic Relations between Words Based on Cooccurrence Patterns in Corpora*. MSc Thesis (in Hebrew). The Hebrew University, Jerusalem, Israel, 2001.

T. Hofmann, J. Puzicha, and. M. Jordan. Learning from dyadic data. In: M.S. Kearns, S.A. Solla and D.A. Cohn, editors. *Advances in Neural Information Processing Systems 11 NIPS*98, pages* 466–472, 1999.

D. Hofstadter and M. Mitchell. The Copycat project: A model of mental fluidity and analogy-making. In D. Hofstadter and the Fluid Analogies Research Group, *Fluid Concepts and Creative Analogies*, pages 205–267, Basic Books, New York, New York, 1995.

J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics COLING-ACL '98*, pages 768–774, Montreal, Canada, 1998.

W. Martin, B. Al and van P. Sterkenburg. On the processing of text corpus: From textual data to lexicographical information. In: R. R. K. Hartmann, editor. *Lexicography: Principles and Practice*, pages 77–87, Academic Press Inc., London, U.K., 1983.

Z. Marx and I. Dagan. Conceptual mapping through keyword coupled clustering. *Mind and Society: A Special Issue on Commonsense and Scientific Reasoning*, 2002, forthcoming.

Z. Marx, I. Dagan and J. M. Buhmann. Coupled Clustering: A method for detecting structural correspondence. In: C. E. Brodley and A. P. Danyluk, editors. *Proceedings of the Eighteenth International Conference on Machine Learning ICML-2001*, pages 353–360, Morgan Kaufmann Publishers, San Francisco, California, 2001.

Z. Marx, I. Dagan and E. Shamir. Detecting sub-topic correspondence through bipartite term clustering. In: A. Kehler and A. Stolke, editors. *Proceedings of the ACL-99 Workshop on Unsupervised Learning in Natural Language Processing*, pages 45–51, College Park, Maryland, 1999.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. *Equation of state calculations by fast computing machines. Journal of Chemical Physics*, 21(6):1087–1092, 1953.

K. J. Nurmela. *Constructing combinatorial designs by local search*. Research report A27. Digital Systems Laboratory, Helsinki University of Technology, Finland, 1993.

M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra and T. Huang. Supporting ranked boolean similarity queries in mars. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):905–925, 1998.

F. C. Pereira, N. Tishby and L. J. Lee. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics ACL' 93*, pages 183–190, Columbus, Ohio, 1993.

J. Puzicha, T. Hofmann and J. M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition* 33(4):617–634, 2000.

H. Schutze. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Minneapolis, Minnesota, 1992.

N. Slonim and N. Tishby. Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, K. R. Müller, editors. *Advances in Neural Information Processing Systems 12 NIPS*99*, pages 617–623, MIT Press, Cambridge, Massachusetts, 2000a.

N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In: N. J. Belkin, P. Ingwersen and M. Leong, editors. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 208–215, Athens, Greece, 2000b.

F. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993.

N. Tishby, F. C. Pereira and W. Bialek. The information bottleneck method. In *37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–379, Urbana-Champaign, Illinois, 1999.

A. Zanasi. Competitive intelligence through data mining public sources. *Competitive Intelligence Review*, 9(1):44–54, 1998.