Chapter 3: Setting and Evaluation

In this chapter, we start laying the grounds to our conception of how to adapt the data-clustering framework, reviewed in the first part of the previous chapter, to the problem of drawing analogies between distinct systems – a problem that is illustratively discussed in the second part of the previous chapter. This chapter describes the basic setting. It explains how the systems to be compared, which are not necessarily similar to one another, are represented within our extended framework and what sorts of clusters are interpretable as conveying analogies or correspondences between the analogized systems. The chapter continues with a preliminary example of an application to real-world textual data of the type treated in depth in the next chapters.

In the last part of this chapter we describe how, given a configuration of clusters of the appropriate kind, the quality of the analogy or the correspondence being drawn is to be evaluated. The evaluation methods are essentially the same ones used to evaluate standard data clustering, but our extended framework suggests some subtle distinctions from the standard framework.

3.1 Problem Setting

The problem examined in this work extends the standard single-set data-clustering problem. In distinction from the setting in the single-set problem, the data for the extended problem is pre-divided into several distinct subsets of elements to be clustered. A setting of two subsets is studied first (Chapter 4). More general setting, which allows a larger number of subsets, is examined later (Chapter 5). Each one of the subsets represents one of two or more systems between which we draw an analogy or a correspondence.

A correspondence between the given subsets is established by means of partitioning them to corresponding partitions. We term each one of the subset parts that result from these partitions a *sub-cluster*. Every one of the obtained sub-clusters has a matching sub-cluster in the other subset (or several matches, one in each subset, in case the data is pre-divided to more than two subsets). Hence, a one-to-one map is established between the sub-clusters of one subset and those of the other subsets. In a setting restricted to two pre-given subsets, a pair of matched sub-clusters is termed a *coupled cluster*. A configuration of an element set pre-divided to two subsets and partitioned into three coupled clusters is sketched in Figure 3.1. Later, when a larger number of pre-given subsets is allowed, a more general term, *cross-partition cluster* will be employed to denote a collection of matched sub-clusters, one from each of the subsets. As a rule, we use the more general latter term, unless the setting under discussion is clearly of the type restricted to two subsets.

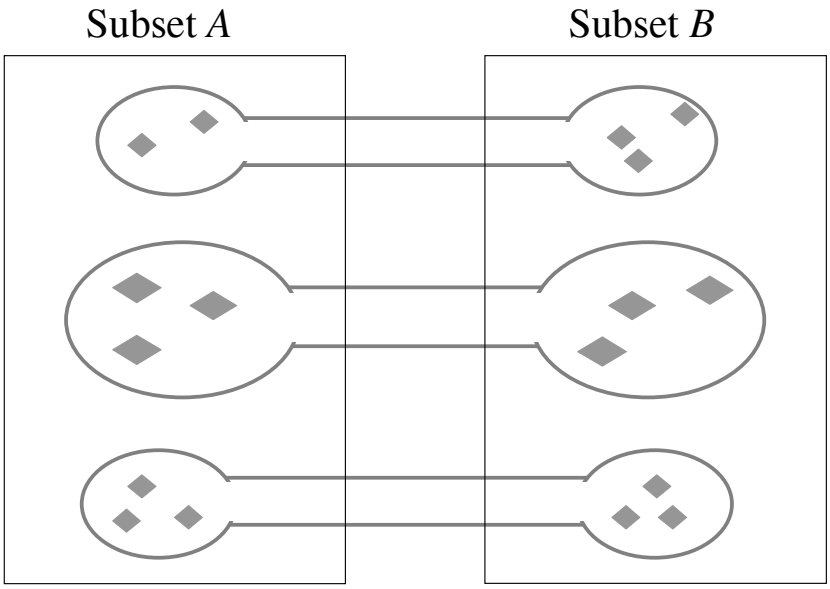

Figure 3.1: An example of a coupled-clustering configuration. The diamonds represent elements of the two pre-given subsets *A* and *B*. Closed contours represent coupled clusters, each of which links two corresponding sub-clusters each from a different subset.

Similarly to the single-set clustering problem discussed in the previous chapter (and many other problems likewise), obtaining a good solution to the cross-partition clustering task, or determining whether a given solution is satisfactory or not, is a matter of optimization over an array of potentially contradicting biases. Standard clustering aims at homogeneous subsets: each cluster is expected to consist of elements that are similar to one another (as much as possible) and, at the same time, its elements are expected to be not similar to elements in other clusters. In the case of coupled and cross-partition clustering, a new requirement is added. On one hand, we still aim at getting homogenous groups of elements. On the other hand, we want to ignore the impact of specificities characterizing any particular pre-given subset. Rather, we require that each homogenous group of elements extracted from one of the subsets would also have a good match – a corresponding group of similar elements – in the other subset or subsets, so that a cross-partition cluster is formed. An optimal configuration would thus consist of clusters containing elements that are similar to one another and distinct from elements in other clusters, subject to the context imposed by the requirement to match sub-clusters from the different subsets.

Similarly to standard-clustering formulations, the computational methods addressing the cross-partition clustering task – including the ones developed in the next chapters – might encounter various sorts of difficulties that are related to the ill-posed nature of the problem (see previous chapter, Subsection 2.1.1.2). In this respect, the situation is not improved in the cross-partition case, as this

task introduces yet another source of potential biases that might not be satisfied in full, additionally to the biases already present in the original data clustering problem. The combination of considerations and biases discussed above defines cross partition clustering as a new task that cannot be tackled through previously studied methods and, particularly, not by standard data clustering methods as explained below.

Technically, it is straightforward to generate a cross-partition clustering configuration by simply ignoring the given pre-partition to subsets and applying a standard clustering method to their union. The current work focuses on cases where a standard single-set clustering method *cannot* work well. Such cases are characterized by relatively homogenous subsets, where the similarity between elements from the same subset is overall higher than similarity between elements originating in different sets. Standard clustering is committed solely to producing homogenous clusters. Therefore, a standard clustering method might tend to produce clusters that coincide with the pre-given subsets or, in case larger number of clusters is requested, clusters that are restricted to elements of an individual subset.

As mentioned in the previous chapter, the case where the pre-given subsets are relatively homogenous but, overall, are not very similar to one another is interesting, as it is characteristic of analogy making. Particularly in such cases, a solution consisting of clusters that are exclusive to one of the pre-given subsets would not reveal correspondence between the subsets. To prevent this non-favorable type of solution to the cross partition clustering problem, our method will be required to include representatives from all subsets in every cluster, along with the basic direction of including similar elements in a cluster. This additional requirement to create clusters that cut across the pre-given partition, while neutralizing regularities internal to specific subsets, differentiates the cross partition clustering problem introduced in this work from the standard data clustering problem.

3.2 A Real-world Example

In principle, cross-partition clustering seems to be applicable to revealing corresponding element groups across any unstructured set of elements pre-divided to several subsets, regardless of the type of data. Hypothetical applicative uses might include: aligning corresponding collections of atomic image components, such as pixels or contours, in order to identify corresponding objects; revealing matches between sets of physiological or psychological records in order to identify equivalencies across distinct subjects or populations and so on.

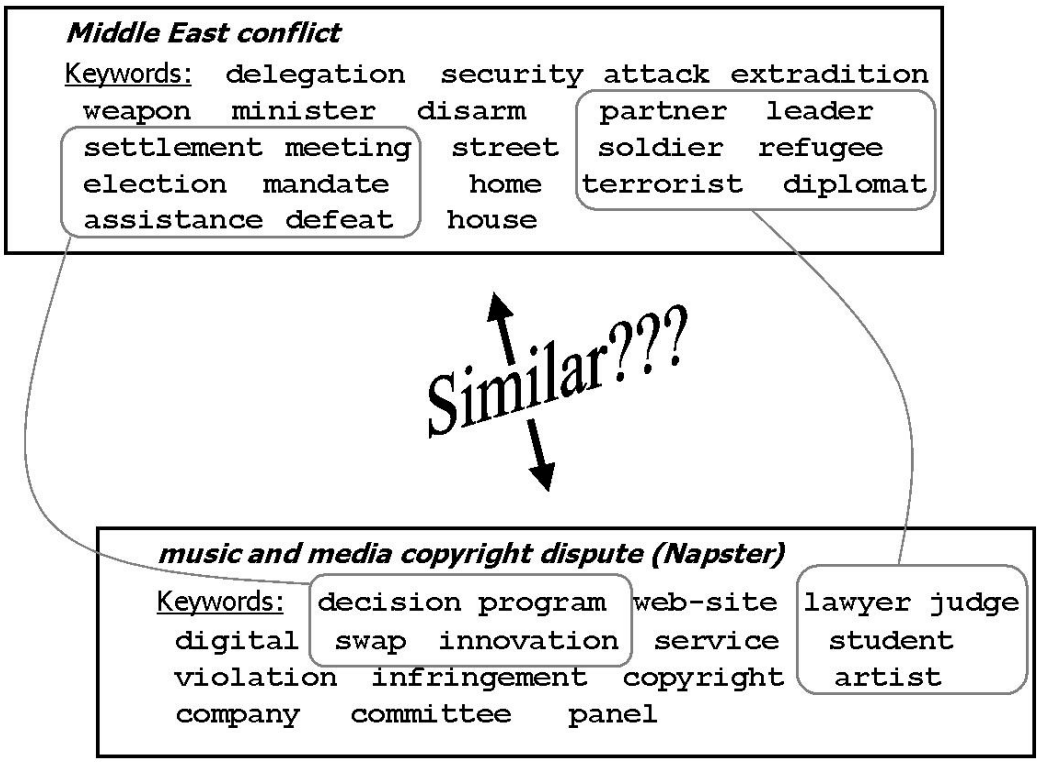

Figure 3.2: Keyword samples from news articles regarding two conflicts. Examples of coupled clusters, each consisting of two matched topical sub-clusters, are marked by curved contours.

The current work applies cross-partition clustering to another task: identification of corresponding topics across texts. Specifically, we have applied the coupled clustering and the cross partition clustering methods to collections of documents containing information regarding distinct domains. The target is to identify prominent sub-topics, themes and categories for which a correspondence can be drawn across the domains. Each domain is characterized by its own terminology and key-concepts extracted from an appropriate corpus. The keyword sets in Figure 3.2, 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 3.2 demonstrates 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.

3.3 Evaluation

Given the ill-posed nature of the standard data-clustering problem, quantitative assessment of the performance of clustering methods is known to be a subtle issue. As the cross partition task is inherently more complex than the standard singe set task, it is at least as problematic to evaluate. The output configuration is expected to balance different types of potentially opposing biases, as discussed in the first section of this chapter, so that the judged quality of the balance obtained might be subjective or application-dependent, just as the case is with solutions to the basic clustering problem.

In general, there are two main strategies for evaluating the quality of a clustering configuration: *internal* criteria and *external* measures. An internal criterion relies solely on the processed data. Using an internal criterion can be an obvious choice when there is a known objective function exactly articulating, in a definite unquestionable manner, what is expected from the clustering mechanism. For many if not most applications, this prerequisite is not met. Our experience with cost functions (see Chapters 4 and 5) particularly demonstrates that cost scores do not capture in general the relative quality of cross-partition clustering configurations with different numbers of clusters. Likewise, relatively small cost differences of configurations resulting from a large similarity matrix often do not reflect differences in the applicative utility of the assessed configurations, even in comparing configurations of equal number of clusters. And, as implied by the ill-posed nature of the task, there are in general several alternative cost functions that may apply to the same data and there is no general procedure to determine which one is the "right" one (also, if we knew the ultimate cost function, we would direct our method to optimize it). Therefore, in this work evaluation of the results is carried out through external measures.

An external evaluation criterion assesses the results relatively to some external "gold standard" – a given configuration $\boldsymbol{E}$ from an authoritative source, assumed to provide the correct solution to the problem. We term the "gold clusters" $e_1, e_2, \ldots, e_l$ that form the criterion configuration $\boldsymbol{E}$, *classes*, to distinguish them from the automatically generated clusters $c_1, c_2, \ldots, c_k$, forming the configuration $\boldsymbol{C}$ that we wish to evaluate.

In our experiments with synthetic data (Sections 4.3 and 5.4.1), the gold standard is given by construction: each data element is drawn as part of some pre-defined class. To evaluate these experiments, we formulate in the following subsection a rather simple and straightforward external criterion, named *purity*.

In many real-world clustering problems, for example topical clustering of keywords, it is not as clear-cut in advance where each element should belong. In such cases (including the present work, see Subsection 4.4.2.3), human judges are often requested to produce the criterion set for evaluation,

based on their judgment and knowledge. Depending on factors such as the specific data type and content and the level of expertise of the human judge, classes produced by human judges might turn to be just a rough criterion for evaluation (so the term "gold standard" might be a bit misleading). In the lack of precise criterion, we collected subjective criterion classes from several participants, so that the level and scope of agreement between them can be inspected as well. We have evaluated those more subtle cases through *Jaccard coefficient* described in Subsection 3.3.2 (several additional external evaluation methods are reviewed and analyzed by Meila, 2003).

3.3.1 Cluster Purity

One straightforward method for measuring the quality of a clustering configuration $\boldsymbol{C}$ in comparison to an external criterion $\boldsymbol{E}$ is known as cluster *purity*. Purity considers all elements of a cluster c in $\boldsymbol{C}$ as if they are classified as members of c's *dominant* class, which is the class e in $\boldsymbol{E}$ with which c shares maximal number of elements. For an individual cluster c, purity is defined as the ratio between those elements shared by c and e, to the total number of elements in c:

$$PURITY_{\boldsymbol{E}}(c) = \frac{1}{|c|} max_{e \in \boldsymbol{E}} \{|c \cap e|\}, \tag{3.1}$$

where $|c \cap e|$ is the number of elements shared by c and e, and $|c|$ is the total number of elements in c. Note that some classes may not share maximal number of elements with any cluster and, complimentarily, several different clusters may share the maximal intersection with the same class.

To evaluate the entire clustering configuration $\boldsymbol{C}$, given the class configuration $\boldsymbol{E}$, compute the average of the cluster-wise purities weighted by the cluster size, which sums up to:

$$PURITY_{\boldsymbol{E}}(\boldsymbol{C}) = \frac{1}{N} \sum_{c \in \boldsymbol{C}} max_{e \in \boldsymbol{E}} \{|c \cap e|\}, \tag{3.2}$$

where N is the total number of data elements.

Purity is a reliable evaluation measure under certain conditions. We use it wherever the criterion is definite and the target number of clusters is known. When it is known that the classes of $\boldsymbol{E}$ provide just a rough approximation of the desired outcome rather than a definite solution, there are several subtleties to consider and more appropriate methods. For instance, incrementing the number of clusters would tend to improve purity, up to the perfect purity of the non-informative partition to singletons. Hence, if the criterion at hand is only an approximation, one might prefer not to restrict the produced output to configurations with number of clusters identical to the number of classes in $\boldsymbol{E}$, neither to commit to any other fixed number of clusters. As these considerations are relevant to our actual experiments and, particularly, we study problems where the number of clusters is not known in advance, we evaluate our results with *Jaccard coefficient*.

3.3.2 Jaccard coefficient

Jaccard coefficient is one of several methods based on element-pair counting (used for evaluating data clustering results also by Ben-Dor, Shamir, & Yakhini, 1999). It symmetrically captures the agreement between an evaluated clustering configuration $\boldsymbol{C}$ and an external classification $\boldsymbol{E}$, on assigning pairs of data elements to the same cluster versus different clusters. A noticeable advantage of the Jaccard coefficient on other pair count method is that it does *not* incorporate those pairs about which the evaluation criterion and evaluated configuration agree that they should not be included in the same cluster. As Ben-Dor et al. (1999) note, this type of agreement is overstressed as the number of clusters grows. Meila, 2003 suggests an alternative: a set-intersection based criterion with information-theoretic motivation that claim to accommodate well to configurations of varied number of clusters. However, it is not clear whether this suggestion is straightforwardly applicable in our case: the fact that our method is applied to a pre-divided element set and the obtained clusters, which are composed of several sub-clusters, might affect the results in a manner that is not trivial to quantify. Therefore, we stick to the pair-count based Jaccard measure, for which incorporating the cross-partition aspect is simpler.

We first introduce the Jaccard measure for the standard deterministic ("hard") clustering case, where each element is assigned to one, and only one, cluster and one criterion class.

The following 0/1 valued functions, are defined for every pair of data elements x and x':

$$\begin{aligned} \textit{Co-assign}_{\boldsymbol{C}}(x,x') &= 1 \text{ iff there is } c\in\boldsymbol{C} \text{ such that } x,x'\in c \text{ (0 otherwise);} \\ \textit{Co-assign}_{\boldsymbol{E}}(x,x') &= 1 \text{ iff there is } e\in\boldsymbol{E} \text{ such that } x,x'\in e \text{ (0 otherwise).} \end{aligned} \tag{3.3}$$

Now, we define pair counts on which the Jaccard coefficient is based.

$$a_{11} = \sum_{x,\,x'\in\boldsymbol{X}} \min\{\ \textit{Co-assign}_{\boldsymbol{C}}(x,x')\,,\ \textit{Co-assign}_{\boldsymbol{E}}(x,x')\ \} \tag{3.4a}$$

(the number of relevant data element pairs assigned into the same cluster by both $\boldsymbol{E}$ and $\boldsymbol{C}$);

$$a_{01} = \sum_{x,\,x'\in\boldsymbol{X}} \min\{\ 1-\textit{Co-assign}_{\boldsymbol{C}}(x,x')\,,\ \textit{Co-assign}_{\boldsymbol{E}}(x,x')\ \} \tag{3.4b}$$

(the number of pairs that have been assigned into the same cluster by $\boldsymbol{E}$ but not by $\boldsymbol{C}$);

$$a_{10} = \sum_{x,\,x'\in\boldsymbol{X}} \min\{\ \textit{Co-assign}_{\boldsymbol{C}}(x,x')\,,\ 1-\textit{Co-assign}_{\boldsymbol{E}}(x,x')\ \} \tag{3.4c}$$

(the number of pairs that have been assigned into the same cluster by $\boldsymbol{C}$ but not by $\boldsymbol{E}$).

Note that Jaccard coefficient ignores a_{00}, which is the agreement between $\boldsymbol{C}$ and $\boldsymbol{E}$ on those pairs that are *not* included in the same clusters. In general a_{00} becomes non-informatively dominant as the number of classes grows.

Jaccard coefficient is defined as

$$JACCARD_{\boldsymbol{E}}(\boldsymbol{C}) = \frac{a_{11}}{a_{11}+a_{10}+a_{01}}. \qquad (3.5)$$

3.3.2.1 Probabilistic Extension for Jaccard coefficient

In the previous chapter we have mentioned data-clustering methods that produce probabilistic output: each element x is distributed over all clusters, so that the association level of x with a cluster c, denoted $p(c|x)$, satisfies $\sum_{c\in\boldsymbol{C}} p(c|x) = 1$ (see previous chapter, Subsection 2.1.2.2). In order to extend the Jaccard coefficient for probabilistic clustering, we modify the definition of the binary function $Co\text{-}assign_{\boldsymbol{C}}$ (Eq. 3.3). The new variation provides a probabilistic value, between 0 and 1, quantifying the level by which two elements x and x' are assigned the same way by a probabilistic $\boldsymbol{C}$:

$$Co\text{-}prob\text{-}assign_{\boldsymbol{C}}(x,x') = \sum_{c\in\boldsymbol{C}} \min\{p(c|x), p(c|x')\}. \qquad (3.6a)$$

This value is equal to 1 if and only if the distributions over the clusters conditioned on both elements are identical. It coincides with the hard clustering case, whenever $p(c|x)$ and $p(c|x')$ are both 0 or 1.

The same probabilistic setting might apply also within the classification criterion $\boldsymbol{E}$: an element may be known, or approximated, as belonging to several criterion classes with either equal or varying levels of assignment (for example, a keyword might be assigned to several sub-topical keyword classes). It is natural, in such case, to require probabilistic assignment levels $p(e|x)$, so that $\sum_{e\in\boldsymbol{E}} p(e|x) = 1$. We modify accordingly the definition of $Co\text{-}assign_{\boldsymbol{E}}(x,x')$:

$$Co\text{-}prob\text{-}assign_{\boldsymbol{E}}(x,x') = \sum_{e\in\boldsymbol{E}} \min\{p(e|x), p(e|x')\}. \qquad (3.6b)$$

Replacing $Co\text{-}assign_{\boldsymbol{C}}$ and $Co\text{-}assign_{\boldsymbol{E}}$ in the definitions of a_{11}, a_{10} and a_{01} (Eq. 3.4a-c) by the newly defined $Co\text{-}prob\text{-}assign_{\boldsymbol{C}}$ and $Co\text{-}prob\text{-}assign_{\boldsymbol{E}}$, we may use the same definition of $JACCARD_{\boldsymbol{E}}(\boldsymbol{C})$ (Eq. 3.5) to define a new measure, $JACCARD\text{-}PROB_{\boldsymbol{E}}(\boldsymbol{C})$, that is usable for evaluating probabilistic clustering results.

The opportunity of evaluating probabilistic clustering brings to mind the question of what is the actual target of probabilistic clustering: does it intend to expose in detail real ambiguities that are present in the data, or is it just a strategy to approximate a deterministic clustering configuration (eventually, the one where every element x is assigned to the cluster c with highest $p(c|x)$)? Both targets are of course legitimate but exposing true ambiguities is a much bigger challenge, as this implies a much richer search space and therefore "noisier" outcome. In practice, it has indeed turned that our actual probabilistic clustering results (Subsection 5.4.2) were scored slightly lower relatively to the corresponding deterministic configurations, so we concentrated on evaluating the deterministic highest-$p(c|x)$ configurations as a replacement to the raw probabilistic outcome.

In distinction from the possible interpretations of probabilistic clustering, whenever multi-assignments are present in the criterion classes as happened occasionally in our experiments, they cannot be interpreted as related with any deterministic assignments. This is the reason that in spite of resorting to deterministic configurations, we still had to use *JACCARD-PROB* rather then the standard *JACCARD* scores. We did so both when we applied deterministic clustering method (Chapter 4) and when we evaluated the deterministic approximation of a probabilistic outcome (Chapter 5).

3.3.2.2 Adapting Jaccard coefficient for the Cross Partition Setting

We propose also a variation of the Jaccard evaluation score that is specifically adapted to the cross-partition clustering setting. In the cross-partition setting, the data is pre-divided to two or more disjoint subsets. Replace a_{11}, a_{01}, and a_{10} defined above (Eq. 3.4) by corresponding quantities, with sums that include only pairs of elements from distinct subsets:

$$\begin{aligned} a'_{11} &= \sum_{x,\,x' \text{ from distinct subsets}} \textit{Co-assign}_{\boldsymbol{C}}(x,x') \cdot \textit{Co-assign}_{\boldsymbol{E}}(x,x'), \\ a'_{01} &= \sum_{x,\,x' \text{ from distinct subsets}} (1 - \textit{Co-assign}_{\boldsymbol{C}}(x,x')) \cdot \textit{Co-assign}_{\boldsymbol{E}}(x,x'), \\ a'_{10} &= \sum_{x,\,x' \text{ from distinct subsets}} \textit{Co-assign}_{\boldsymbol{C}}(x,x') \cdot (1 - \textit{Co-assign}_{\boldsymbol{E}}(x,x')). \end{aligned} \tag{3.7}$$

Plugging a'_{11}, a'_{11}, a'_{11} into the definition of Jaccard coefficient (Eq. 3.5), we obtain:

$$\textit{JACCARD-CP}_{\boldsymbol{E}}(\boldsymbol{C}) = \frac{a'_{11}}{a'_{11}+a'_{10}+a'_{01}}, \tag{3.8}$$

a variation on Jaccard coefficient adapted to the context of the new problem, which considers only elements from the distinct subsets and excludes the impact of all within-subset pairs. If we also replace in Eq. 3.7 above *Co-assign* by *Co-prob-assign* (Eq. 3.6), we obtain the *JACCARD-PROB-CP* measure, which combines probabilistic clustering with ignoring the impact of the within-subset pairs.

Similarly to the corresponding question regarding evaluation of probabilistic clustering, the question of whether to use *JACCARD-CP* as a replacement for the original *JACCARD* measure depends on how exactly one views the target of the cross-partition task. If the emphasis is on creating a variety of associations between the pre-given subsets, then it would make sense to use *JACCARD-CP* (or *JACCARD-PROB-CP*). If, in subtle distinction, the focus is on revealing concepts or themes that cut across the pre-given partition and might nevertheless incorporate some within-subset information, then using the original *JACCARD* measure (or *JACCARD-PROB*) can be viewed as more appropriate. In evaluating our actual results, we accordingly use *JACCARD-PROB-CP* for evaluating our coupled clustering method (Section 4.4) that, as will be explained, relies solely on cross-subset information. For evaluating the later cross-partition method (Subsection 5.4.2), which does not ignore within-subset information, we use the *JACCARD-PROB* measure.