Cross-dataset Clustering: Revealing Corresponding Themes Across Multiple Corpora

Ido DAGAN
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel, 52900
and LingoMotors Inc.
dagan@lingomotors.com

Zvika MARX
Center for Neural Computation
The Hebrew University
and CS Dept., Bar-Ilan University
Ramat-Gan, Israel, 52900
zvim@cs.huji.ac.il

Eli SHAMIR
School of Computer Science
and Engineering
The Hebrew University
Jerusalem, Israel, 91904
shamir@cs.huji.ac.il

Abstract

We present a method for identifying corresponding themes across several corpora that are focused on related, but distinct, domains. This task is approached through simultaneous clustering of keyword sets extracted from the analyzed corpora. Our algorithm extends the information-bottleneck soft clustering method for a suitable setting consisting of several datasets. Experimentation with topical corpora reveals similar aspects of three distinct religions. The evaluation is by way of comparison to clusters constructed manually by an expert.

1 Introduction

This paper addresses the problem of detecting corresponding subtopics, or themes, within related bodies of text. Such task is typical to comparative research, whether commercial or scientific: a conceivable application would aim at detecting corresponding characteristics regarding, e.g., companies, markets, legal systems or political organizations.

Clustering has often been perceived as a mean for extracting meaningful components from data (Tishby, Pereira and Bialek, 1999). Regarding textual data, clusters of words (Pereira, Tishby and Lee, 1993) or documents (Lee and Seung, 1999; Dhillon and Modha, 2001) are often interpreted as capturing topics or themes that play prominent role in the analyzed texts.

Our work extends the "standard" clustering paradigm, which pertains to a single dataset. We address a setting in which several datasets, corresponding to related domains, are given. We focus on the comparative task of detecting those themes that are expressed across several datasets, rather than discovering internal themes within each individual dataset.

More specifically, we address the task of clustering simultaneously multiple datasets such that each cluster includes elements from several datasets, capturing a common theme, which is shared across the sets. We term this task *cross-dataset* (CD) clustering.

In this article we demonstrate CD clustering through detecting corresponding themes across three different religions. That is: we apply our approach to three sets of religion-related keywords, extracted from three corpora, which include encyclopedic entries and introductory articles regarding Buddhism, Christianity and Islam. Each one of three representative keyword-sets, which were extracted from the above corpora, presumably encapsulates topics and themes discussed within its source corpus. Our algorithm succeeds to reveal common themes such as *scriptures*, *rituals* and *schools* through respective keyword clusters consisting of terms such as **Sutra**, **Bible** and **Quran**; **Full Moon**, **Easter** and **Id al Fitr**; **Theravada**, **Protestant** and **Shiite** (see Table 1 below for a detailed depiction of our results).

The CD clustering algorithm, presented in Section 2.2 below, extends the *information bottleneck* (IB) soft clustering method. Our modifications to the IB formulation enable the clustering algorithm to capture characteristic patterns that run across different datasets, rather then being "trapped" by unique characteristics of individual datasets.

Like other topic discovery tasks that are approached by clustering, the goal of CD clustering is not defined in precise terms. Yet, it is clear that its focus on detecting themes in a comparative manner, within multiple datasets, distinguishes CD clustering substantially from the standard single-dataset clustering paradigm.

A related problem, of detecting analogies between different information systems has been addressed in the past within cognitive research (e.g. Gentner, 1983; Hofstadter et al., 1995). Recently, a related computational method for detecting corresponding themes has been introduced (*coupled clustering*, Marx et al., 2002). The coupled clustering setting, however, being focused on detecting analogies, is limited to two data sets. Further, it requires similarity values between pairs of data elements as input: this setting does not seem straightforwardly applicable to the multiple dataset setting. Our method, in distinction, uses a more direct source of information, namely word co-occurrence statistics within the analyzed corpora. Another difference is that we take the "soft" approach to clustering, producing probabilities of assignments into clusters rather than a deterministic 0/1 assignment values.

2 Algorithmic Framework

2.1 Review of the IB Clustering Algorithm

The information bottleneck (IB) iterative clustering method is a recent approach to soft (probabilistic) clustering for a single set, denoted by X, consisting of elements to be clustered (Tishby, Pereira & Bialek, 1999). Each element $x \in X$ is identified by a probabilistic feature vector, with an entry, $p(y|x)$, for every feature y from a pre-determined set of features Y. The $p(y|x)$ values are estimated from given co-occurrence data:

$$p(y|x) = \frac{count(x,y)}{\sum_{y' \in Y} count(x,y')}$$

(hence $\sum_{y \in Y} p(y|x) = 1$ for every x in X).

The IB algorithm is derived from information theoretic considerations that we do not address here. It computes, through an iterative EM-like process, probabilistic assignments $p(c|x)$ for each element x into each cluster c. Starting with random (or heuristically chosen) $p(c|x)$ values at time $t = 0$, the IB algorithm iterates the following steps until convergence:

IB1: Calculate for each cluster c its marginal probability:

$$p_t(c) = \sum_{x \in X} p(x) p_{t-1}(c|x) .$$

IB2: Calculate for each feature y and cluster c a conditional probability $p(y|c)$:

$$p_t(y|c) = \sum_{x \in X} p(y|x) p_{t-1}(x|c) .$$

(Bayes' rule is used to compute $p(x|c)$).

IB3: Calculate for each element x and each cluster c a value $p(c|x)$, indicating the "probability of assignment" of x into c:

$$p_t(c|x) = \frac{p_t(c) sim_t^{Y,\beta}(x,c)}{\sum_{c'} p_t(c') sim_t^{Y,\beta}(x,c')} ,$$

with $sim_t^{Y,\beta}(x,c) = \exp\{-\beta D_{KL}[p(y|x) \| p_t(y|c)]\}$ (D_{KL} is the *Kullback-Leibler divergence*, see Cover & Thomas, 1991).

The parameter β controls the sensitivity of the clustering procedure to differences between the $p(y|c)$ values. The higher β is, the more "determined" the algorithm becomes in assigning each element into the closest cluster. As β is increased, more clusters that are separable from each other are obtained upon convergence (the target number of clusters is fixed). We want to ensure, however, that assignments do not follow more than necessary minute details of the given data, as a result of too high β (similarly to over generalization in supervised settings). The IB algorithm is therefore applied repeatedly in a cooling-like process: it starts with a low β value, corresponding to low temperature, which is increased every repetition of the whole iterative converging cycle, until the desired number of separate clusters is obtained.

2.2 The Cross-dataset (CD) Clustering Method

The (soft) CD clustering algorithm receives as input multiple datasets along with their feature vectors. In the current application, we have three sets extracted from the corresponding corpora – $X_{Buddhism}$, $X_{Christianity}$, and X_{Islam} – each of ~150 keywords to be clustered. A particular keyword might appear in two or more of the

datasets, but the CD setting considers it as a distinct element within each dataset, thus keeping the sets of clustered elements disjoint. Like the IB clustering algorithm, the CD algorithm produces probabilistic assignments of the data elements.

The feature set Y consists, in the current work, of about 7000 content words, each occurs in at least two of the examined corpora. The set of features is used commonly for *all* datasets, thus it underlies a common representation, which enables the clustering process to compare elements of different sets.

Naively approached, the original IB algorithm could be utilized unaltered to the multiple-dataset setting, simply by applying it to the unified set X, consisting of the union of the disjoint X_i's. The problem of this simplistic approach is that each dataset has its own characteristic features and feature combinations, which correspond to prominent topics discussed uniquely in that corpus. A standard clustering method, such as the IB algorithm, would have a tendency to cluster together elements that originate in the same dataset, producing clusters populated mostly by elements from a single dataset (cf. Marx et al, 2002). The goal of CD clustering is to neutralize this tendency and to create clusters containing elements that share common features across *different* datasets.

To accomplish this goal, we change the criterion by which elements are assigned into clusters. Recall that the assignment of an element x to a cluster c is determined by the similarity of their characterizing feature distributions, $p(y|x)$ and $p(y|c)$ (step IB3). The problem lies in using the $p(y|c)$ distribution, which is determined by summing $p(y|x)$ values over all cluster elements, to characterize a cluster *without* taking into account dataset boundaries. Thus, for a certain y, $p(y|c)$ might be high despite of being characteristic only for cluster elements originating in a single dataset. This results in the tendency discussed above to favor clusters consisting of elements of a single dataset.

Therefore, we define a biased probability distribution, $\tilde{p}^c(y)$, to be used by the CD clustering algorithm for characterizing a cluster c. It is designed to call attention to y's that are typical for cluster members in all, or most, different datasets. Consequently, an element x would be assigned to a cluster c (as in step IB3) in accordance to the degree of similarity between its own characteristic features and those characterizing other cluster members from all datasets. The resulting clusters would thus contain representatives of all datasets.

The definition of $\tilde{p}^c(y)$ is based on the joint probability $p(y,c,X_i)$. First, compute the geometric mean of $p(y,c,X_i)$ over all X_i, weighted by $p(X_i)$:

$$\rho(y,c) = \prod_i (p(y,c,X_i))^{p(X_i)}$$

(see Appendix below for the details of how $p(X_i)$ and $p(y,c,X_i)$ are calculated).

ρ is not a probability measure, but just a function of y and c into [0,1]. However, since a geometric mean reflects "uniformity" of the averaged values, ρ captures the degree to which $p(y,c,X_i)$ values are high across *all* datasets.

We found empirically that at this stage, it is advantageous to normalize ρ across all clusters and then to rescale the resulting probabilities (over the c's, for each y) by the original $p(y)$:

$$\rho'(y,c) = (\rho(y,c) / \sum_{c'} \rho(y,c')) \times p(y) .$$

Finally, to obtain a probability distribution over y for each cluster c, normalize the obtained $\rho'(y,c)$ over all y's:

$$\tilde{p}^c(y) = \rho'(y,c) / \sum_{y'} \rho'(y',c) .$$

As explained, $\tilde{p}^c(y)$ characterizes c (analogously to $p(y|c)$ in IB), while ensuring that the feature-based similarity of c to any element x reflects feature distribution across all data sets.

The CD clustering algorithm, starting at $t = 0$, iterates, in correspondence to the IB algorithm, the following steps:

CD1: Calculate for each cluster c its marginal probability (same as IB1):

$$p_t(c) = \sum_{x \in \bigcup_i X_i} p(x) p_{t-1}(c \mid x) .$$

CD2: Compute $\tilde{p}^c(y)$ as described above.

CD3: Compute $p(c|x)$ (with $\tilde{p}^c(y)$ playing the role played by $p(y|c)$ in IB3):

$$p_t(c \mid x) = \frac{p_t(c) SIM_t^{Y,\beta}(x,c)}{\sum_{c'} p_t(c') SIM_t^{Y,\beta}(x,c')} ,$$

with $SIM_t^{Y,\beta}(x,c) = \exp \{-\beta D_{KL}[p(y|x) \| \tilde{p}_t^c(y)]\}$.

3 CD Clustering for Religion Comparison

The three corpora that are focused on the compared religions – Buddhism, Christianity and Islam – have been downloaded from the Internet. Each corpus contains 20,000–40,000 word tokens (5–10 Megabyte). We have used a text mining tool to extract most religion keywords that form the three sets to which we applied the CD algorithm. The software we have used – TextAnalyst 2.0 – identifies within the corpora key-phrases, from which we have excluded items that appear in fewer than three

Table 1: Results of religion keyword CD clustering. The authors have set the cluster titles. For each cluster *c* and each religion, the 15 keywords *x* with the highest probability of assignment within the cluster are displayed (assignment probabilities, i.e. $p(c|x)$ values are indicated in brackets). Terms that were used by the expert (see Table 2) are underlined.

Buddhism	Christianity	Islam
$\hat{\mathbf{c}}_1$ ***(Cherished Qualities)***		
god (.68), amida (.58), **bodhisattva** (.50), salvation (.45), enlightenment (.43), deva (.43), attain (.41), **sacrifice** (.39), awaken (.25), spirit (.25), **nirvana** (.24), **buddha nature** (.24), humanity (.22), speech (.18), teach (.18)	**god** (.69), good works (.65), **love of god** (.62), salvation (.60), gift (.58), intercession (.56), repentance (.55), righteousness (.53), peace (.52), **love** (.51), obey god (.49), saviour (.48), atonement (.46), holy ghost (.45), **jesus christ** (.45)	god (.86), **one god** (.84), **allah** (.76), bless (.76), worship (.75), submission (.73), peace (.73), command (.72), guide (.71), divinity (.70), messanger (.70), believe (.62), mankind (.61), commandment (.58), witness (.57)
$\hat{\mathbf{c}}_2$ ***(Customs and Festivals)***		
full moon (.99), **stupa** (.98), **mantra** (.96), pilgrim (.96), **monastery** (.89), **temple** (.86), **statue** (.73), worship (.61), **monk** (.54), **mandala** (.32), trained (.23), bhikkhu (.15), disciple (.12), **meditation** (.11), nun (.11)	easter (.99), **sunday** (.99), **christmas** (.99), service (.98), city (.98), **eucharist** (.96), pilgrim (.95), pentecost (.93), **jerusalem** (.91), **pray** (.89), worship (.82), **minister** (.73), ministry (.70), read bible (.50), mass (.24)	**id al fitr** (.99), **friday** (.99), **ramadan** (.99), eid (.99), **pilgrim** (.99), **mosque** (.99), **mecca** (.99), **kaaba** (.99), salat (.99), **fasting** (.99), **medina** (.98), city (.98), **pray** (.98), hijra (.97), **charity** (.96)
$\hat{\mathbf{c}}_3$ ***(Spiritual States)***		
phenomena (.94), problem (.93), mindfulness (.92), awareness (.92), consciousness (.91), law (.88), **emptiness** (.88), **samadhi** (.87), sense (.87), experience (.86), wisdom (.83), moral (.83), karma (.82), find (.81), exist (.80)	moral (.96), problem (.94), argue (.91), question (.87), argument (.74), experience (.73), incarnation (.72), relationship (.71), idolatry (.58), find (.45), law (.41), learn (.38), **confession** (.34), foundation (.32), faith (.31)	moral (.93), spirit (.79), question (.75), life (.71), freedom (.67), existence (.56), humanity (.53), find (.52), faith (.52), code (.51), law (.41), universe (.39), being (.36), teach (.35), commandment (.29)
$\hat{\mathbf{c}}_4$ ***(Sorrow, Sin, Punishment and Reward)***		
lamentation (.99), grief (.99), animal (.89), pain (.87), death (.86), kill (.84), **reincarnation** (.81), realm (.76), samsara (.69), rebirth (.61), **dukkha** (.56), anger (.53), soul (.43), **nirvana** (.43), birth (.33)	punish (.94), **hell** (.93), violence (.86), **fish** (.86), sin (.83), earth (.81), soul (.78), death (.77), sinner (.76), sinful (.74), **heaven** (.73), **satan** (.72), **suffer** (.71), flesh (.71), judgment (.67)	**hell** (.97), earth (.88), **heaven** (.87), death (.85), sin (.85), alcohol (.69), satan (.60), face (.59), day of judgment (.52), deed (.48), angel (.25), being (.24), universe (.16), existence (.13), bearing (.12)
$\hat{\mathbf{c}}_5$ ***(Schools, Traditions and their Originating Places)***		
korea (.99), china (.99), tibet (.99), theravada (.99), **school** (.99), asia (.99), founded (.99), west (.99), sri lanka (.99), mahayana (.99), india (.99), history (.99), hindu (.99), japan (.99), study (.99)	**cardinal** (.99), orthodox (.99), protestant (.99), **university** (.99), **vatican** (.99), catholic (.99), **bishop** (.99), **rome** (.99), **pope** (.99), **monk** (.99), tradition (.99), **theology** (.99), baptist (.98), **church** (.98), **divinity** (.93)	africa (.99), shiite (.99), sunni (.99), **shia** (.99), west (.99), christianity (.99), arab (.99), founded (.98), arabia (.97), **sufi** (.96), history (.96), fiqh (.95), scholar (.91), **imam** (.90), jew (.89)
$\hat{\mathbf{c}}_6$ ***(Names, Places, Characters, Narratives)***		
gautama (.96), king (.95), friend (.68), disciple (.60), birth (.48), hear (.43), ascetic (.41), amida (.40), deva (.33), teach (.19), **sacrifice** (.15), statue (.14), **buddha** (.12), **bodhisattva** (.12), **dharma** (.09)	**bethlehem** (.98), **jordan** (.97), **mary** (.95), lamb (.90), king (.90), **second coming** (.81), born (.76), israel (.74), child (.73), elijah (.72), baptize (.70), **john the baptist** (.68), **priest** (.68), adultery (.65), **zion** (.61)	husband (.99), ismail (.98), father (.97), son (.95), mother (.94), born (.92), wife (.92), child (.89), **ali** (.88), musa (.71), isa (.70), ibrahim (.67), **caliph** (.43), tribe (.35), saint (.30)
$\hat{\mathbf{c}}_7$ ***(Scripture)***		
tripitaka (.98), sanskrit (.94), translate (.93), **sutra** (.85), discourse (.79), **pali canon** (.74), story (.66), book (.64), word (.61), write (.45), **buddha** (.39), **lama** (.32), text (.23), **dharma** (.21), teacher (.17)	hebrew (.99), translate (.99), gospels (.99), greek (.99), book (.98), **new testament** (.98), **old testament** (.96), passage (.96), **matthew** (.95), write (.94), **luke** (.93), **apostle** (.93), **bible** (.91), **paul** (.90), **john** (.90)	translatee (.99), bible (.99), write (.98), book (.97), **hadith** (.96), **sunna** (.96), **quran** (.94), word (.93), story (.93), revelation (.88), companion (.80), **muhammad** (.80), **prophet** (.73), writing (.71), read quran (.46)

corpus documents[1]. Thus, composite and rare terms as well as phrases that the software has inappropriately segmented were filtered out. We have added to the automatically extracted terms additional items contributed by a comparative religion expert (about 15% of the sets were thus not extracted automatically, but those terms occur frequently enough to underlie informative co-occurrence vectors).

The common set of features consists of all corpus words that occur in at least three different documents within two or three of the corpora, excluding a list of common function words. Co-occurrences were counted within a bi-directional five-word window, truncated by sentence ends.

The number of clusters produced – seven – was empirically determined as the maximal number with relatively large proportion ($p(c) > .01$) for all clusters. Trying eight clusters or more, we obtain clusters of minute size, which apparently do not reveal additional themes or topics. Table 1 presents, for each cluster c and each religion, the 15 keywords x with the highest $p(c|x)$ values. The number 15 has no special meaning other than providing rich, balanced and displayable notion of all clusters. The displayed 3×15 keyword subsets are denoted $\hat{c}_1 \ldots \hat{c}_7$.

We gave each cluster a title, reflecting our (naive) impression of its content. As we interpret the clusters, they indeed reveal prominent aspects of religion: rituals ($\hat{c}_2$), schools ($\hat{c}_5$), narratives ($\hat{c}_6$) and scriptures ($\hat{c}_7$). More delicate issues, such as cherished qualities ($\hat{c}_1$), spiritual states ($\hat{c}_3$), suffering and sin ($\hat{c}_4$) are reflected as well, in spite of the very different position taken by the distinct religions with regard to these issues.

3.1 Comparison to Expert Data

We have asked an expert of comparative religion studies to simulate roughly the CD clustering task: assigning (freely-chosen) keywords into corresponding subsets, reflecting prominent resembling aspects that cut across the three examined religions. The expert was not asked to indicate a probability of assignment, but he was allowed to use the same keyword in more than one cluster. The expert clusters, with the exclusion of terms that we were not able to locate in our corpora, are displayed in Table 2. In addition to our tags – $e_1 \ldots e_8$ – the expert gave a title to each cluster.

Although the keyword-clustering task is highly subjective, there are notable overlapped regions shared by the expert clusters and ours. Two expert topics – 'Books' (e_1) and 'Ritual' (e_3) – are clearly captured (by $\hat{c}_7$ and $\hat{c}_2$ respectively). 'Society and Politics' (e_4) and 'Establishments' (e_5) – are both in some correspondence with our 'Schools and Traditions' cluster ($\hat{c}_5$). On the other hand, our output fails to capture the 'Mysticism' expert cluster (e_6). Further, our output suggests the 'spiritual states' theme ($\hat{c}_3$) and distinguishes cherished qualities ($\hat{c}_1$) from sin and suffering ($\hat{c}_4$). Such observations might make sense but are not covered by the expert.

To quantify the overall level of overlap between our output and the expert's, we introduce suitable versions of *recall* and *precision* measures.

We want the recall measure to reflect the proportion of expert terms that are captured by our configuration, provided that an optimal correspondence between our clusters to the expert is considered. Hence, for each expert cluster, e_j, we find a particular $\hat{c}_k$ with maximal number of overlapping terms (note that two or more expert clusters are allowed to be covered by a single $\hat{c}_k$, to reflect cases where several related sub-topics are merged within our results). Denote this maximal number by $M(e_j)$:

$$M(e_j) = \max_{k=1\ldots7} |\{x \in e_j : x \in \hat{c}_k\}| .$$

Consequently, the recall measure R is defined to be the sum of the above maximal overlap counts over all expert clusters, divided by all 131 expert terms (repetitions in distinct clusters counted):

$$R = \sum_{j=1\ldots8} M(e_j) / 131.$$

To estimate how precise our results are, we are interested in the relative part of our clusters, reduced to the expert terms, which has been assigned to the "right" expert cluster by the same optimal correspondence. Note that in this case we do not want to sum up several M values that are associated with a single $\hat{c}_k$: a single cluster covering several expert clusters should be considered as an indication of poor precision. Furthermore, if we do this, we might recount

[1] An evaluation copy of TextAnalyst, by MicroSystems Ltd., can be downloaded from http://www.megaputer.com/php/eval.php3

some of $\hat{c}_k$'s terms (specifically, keywords that the expert has included in several clusters; this might result in precision > 100%). We need therefore to consider at most one M value per $\hat{c}_k$, namely the largest one. Define

$$M^*(\hat{c}_k) = \max_{j=1\ldots7} \{M(e_j): |\hat{c}_k \cap e_j| = M(e_j)\}$$

($M^*(\hat{c}_k) = 0$ if the set on the right-hand side is empty, i.e. there is no e_j that share $M(e_j)$ elements with $\hat{c}_k$). The global precision measure P is the sum of all M^* values, divided by the number of expert terms appearing among the $\hat{c}_k$'s (repetitions counted), which are, in the current case, the 94 underlined terms in Table 1:

$$P = \sum_{k=1\ldots7} M^*(\hat{c}_k) / 94.$$

Our algorithm has achieved the following values: R = 67/131 = 0.51, P = 58/94 = 0.62. This is a notable improvement relatively to the IB algorithm results: R = 42/131 = 0.32 and P = 32/82 = 0.39 (random assignment of the keywords into seven clusters yield average values R = 0.36, P = 0.33). As we have expected, three of the clusters produced by the IB algorithms are populated, with very high probability, by most keywords of a single religion. Within these specific religion clusters as well as the other sparsely populated clusters, the ranking inducted by the $p(c|x)$ values is not very informative regarding particular sub-topics. Thus, the IB performs the CD clustering task poorly, even in comparison to random results.

We note that, similarly to our algorithm, the IB algorithm produces at most 7 clusters of non-negligible size. This somewhat supports our

Table 2: The expert cross-dataset clusters. Cluster titles were assigned by the expert. For each expert cluster, the best fitting automated cross-dataset cluster is indicated on the right-hand side, as well as the number of relevant expert words it includes. The terms of this best-fit cluster are underlined. Superscripts indicate indices of the cross-dataset cluster(s), among $\hat{c}_1 \ldots \hat{c}_7$, to which each term belongs.

Buddhism	Christianity	Islam
e_1: *Scriptures*		**$\hat{C}_7$ → 14 (of 19)**
sutra[7], mantra[2], mandala[2], koan, pali canon[7]	new testament[7], old testament[7], bible[7], apostle[7], revelation, john[7], paul[7], luke[7], matthew[7]	quran[7], hadith[7], sunna[7], sharia, muhammad[7]
e_2: *Beliefs and Ideas*		**$\hat{C}_4$ → 10 (of 25)**
nirvana[14], four noble truths, dharma[6,7], dukkha[4], buddha nature[1], tantra, emptiness[3], reincarnation[4]	resurrection, heaven[4], hell[4], trinity, second coming[6], jesus christ[1], love of god[1], god[1], satan[4], cross, dove[4], fish[4]	prophet[7], allah[1], one god[1], five pillars, heaven[4], hell[4]
e_3: *Ritual, Prayer, Holydays*		**$\hat{C}_2$ → 16 (of 20)**
meditation[2], statue[2], sacrifice[1,6], gift, stupa[2]	sunday[2], pray[2], confession[3], eucharist[2], christmas[2], baptism	pilgrim[2], charity[2], ramadan[2], fasting[2], id al fitr[2], pray[2], friday[2], kaaba[2], mecca[2]
e_4: *Society and Politics*		**$\hat{C}_5$ → 9 (of 19)**
dalai lama, monk[2], bodhisattva[1,6], lama[7]	rome[5], vatican[5], church[5], minister[2], priest[6], cardinal[5], pope[5], bishop[5]	sharia, caliph[6], imam[5], shia[5], sunna[7], ali[6], sufi[5]
e_5: *Establishments*		**$\hat{C}_5$ → 6 (of 10)**
monastery[2], temple[2], sangha, school[5]	church[5], cardinal[5], pope[5], bishop[5]	mosque[2], imam[5]
e_6: *Mysticism*		**$\hat{C}_2$ → 2 (of 11)**
meditation[2], nirvana[14], samadhi[3], tantra	eucharist[2], miracle, crucifixion, suffer[4], love[1], saint	sufi[5]
e_7: *Learning and Education*		**$\hat{C}_5$ → 4 (of 8)**
monastery[2], monk[2], sutra[7], meditation[2]	monk[5], university[5], theology[5], divinity[5]	
e_8: *Names and Places*		**$\hat{C}_6$ → 7 (of 20)**
gautama[6], buddha[6,7]	jesus christ[1], john the baptist[6], jordan[6], jerusalem[2], bethlehem[6], mary[6], rome[5], john[7], paul[7], luke[7], matthew[7], zion[6]	muhammad[7], ali[6], mecca[2], medina[2]

impression that the limit on number of "interesting" clusters reflects intrinsic exhaustion of the information embodied within the given data. It is yet to be carefully examined whether this observation provides any hint regarding the general issue of the "right" number of clusters.

4 Conclusion

This paper addressed the relatively unexplored problem of detecting corresponding themes across multiple corpora. We have developed an extended clustering algorithm that is based on the appealing and highly general Information Bottleneck method. Substantial effort has been devoted to adopting this method for the Cross-Dataset clustering task.

Our approach was demonstrated empirically on the challenging task of finding corresponding themes across different religions. Subjective examination of the system's output, as well as its comparison to the output of a human expert, demonstrate the potential benefits of applying this approach in the framework of comparative research, and possibly in additional text mining applications.

Given the early stage of this line of research, there is plenty of room for future work. In particular, further research is needed to provide theoretic grounding for the CD clustering formulations and to specify their properties. Empirical work is needed to explore the potential of the proposed paradigm for other textual domains as well as for related applications. Particularly, we have recently presented a similar framework for template induction in information extraction (*cross-component clustering*, Marx, Dagan, & Shamir, 2002), which should be studied in relation to the CD algorithm presented here.

Appendix

The value of $p(X_i)$, which is required for the calculations in Section 3.2, is given directly from the input co-occurrence data as follows:

$$p(X_i) = \frac{\sum_{x\in X_i\ y\in Y} count(x,y)}{\sum_{x'\in X\ y\in Y} count(x',y)}$$

The values $p_t(c|X_i)$, $p_t(y|c,X_i)$ are calculated from values that are available at time step $t-1$:

$$p_t(c \mid X_i) = \sum_{x\in X_i} p(x) p_{t-1}(c \mid x),$$

$$p_t(y \mid c, X_i) = \sum_{x\in X_i} p(y \mid x) p_{t-1}(x \mid c, X_i)$$

($p_{t-1}(x|c,X_i)$ is due to Bayes' rule conditioned on X_i: $p_{t-1}(x|c,X_i) = p_{t-1}(c|x) \times p(x) / p_{t-1}(c|X_i)$; note that $p_{t-1}(c|x) = p_{t-1}(c|x,X_i)$).

Finally we have:

$$p_t(y,c,X_i) = p_t(y|c,X_i) \times p_t(c|X_i) \times p(X_i).$$

Acknowledgments

We thank Eitan Reich for providing the expert data, as well as for illuminating discussions.

This work has been partially supported by ISRAEL SCIENCE FOUNDATION founded by The Academy of Sciences and Humanities (grants 574/98-1 and 489/00).

References

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

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

Gentner, D. (1983). Structure-mapping: a theoretical framework for analogy. *Cognitive Science*, *7*, pp. 155–170.

Hofstadter, D. R. and the Fluid Analogies Research Group (1995). *Fluid Concepts and Creative Analogies*. New-York: Basic Books, 518 p.

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

Marx, Z., Dagan, I. and Shamir, E. (2002). Cross-component clustering for template induction. Workshop on Text Learning (TextML-2002), Sydney, Australia.

Marx, Z., Dagan, I., Buhmann, J. M. and Shamir, E. (2002). Coupled clustering: a method for detecting structural correspondence. *Journal of Machine Learning Research*, accepted for publication.

Pereira, F. C. N., Tishby N. and Lee L. J. (1993). Distributional clustering of English words. In: *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics ACL' 93*, Columbus, OH, pp. 183–190.

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