Cross-component Clustering for Template Induction

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

Ido Dagan DAGAN@LINGOMOTORS.COM
Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, ISRAEL

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

Abstract

We suggest an unsupervised approach to template induction for information extraction, through detecting sub-topics and themes that cut across the documents of a topical corpus. We introduce a new method – cross component clustering – that simultaneously clusters the components forming our setting, each of which consists of the words of a single article. Our algorithm is derived from the Information Bottleneck clustering algorithm. The resulting clusters are found to be in systematic correspondence with sets of terms that are used in filling the slots of the MUC3/4 ready-made template, which was used for evaluation.

1. Introduction

This paper introduces an unsupervised approach to support inducing templates for directing systematic extraction of information from documents (*Information Extraction* – IE; see, e.g. Gaizauskas & Wilks, 1998). Defining a template typically depends on the users' perspectives, goals and interests. Still, an unsupervised pre-processing procedure, which would function complementarily to available domain-specific guidelines, could provide a valuable aid. The apparent advantage of such procedure lies, in fact, in its objectivity and the straightforward adaptation to any arbitrary domain.

The Information Bottleneck (IB) clustering method, on which our algorithmic approach is based, interprets clustering as extracting meaningful factors from the clustered data (Tishby, Pereira and Bialek, 1999). Clustering of words, in particular, has been applied for revealing concepts, sub-topics and themes that are prominent in the examined corpus (Pereira, Tishby and Lee, 1993). The IB framework, which is presented in detail in Section 2.1 below, is based on co-occurrence counts. Words, which are the clustered elements in this work, are clustered according to their adjacent co-occurrences in text with other words that are members in a set designated as the set of *features*. Basically, given a pair of clustered words, the more common co-occurrences with features they share, higher are the chances that they are assigned into the same cluster. Section 3.1 below characterizes in detail the particular words that are being clustered well as the feature words being used.

Our approach is demonstrated on a well-studied dataset – the MUC3/4 corpus[1] – consisting of news articles reporting terror events that took place in South America during the 80's (Chinchor, Hirschman & Lewis, 1993). In addition to the corpus articles, the MUC3/4 dataset includes a ready-made IE template, with respect to which we evaluate our results. Additional MUC3/4 files (the *training files*) detail the exact terms or phrases with which each one of the template slots is supposed to be filled, with regard to every relevant article. For example: the slot that captures the location in which the event took place, should be filled with the phrase Ecuador: Quito (city) for one article and with Itagui (municipality) for another article. The slot that refers to the description of the event's human target is supposed to be filled with phrases such as leftist presidential candidate, party leader, or Jesuit priest (see Subsection 3.2, Table 2 for more examples).

The present work aims at revealing, through clustering, the themes shared by collections of slot-related phrases, such as those exemplified above. We expect the sets of words occurring in the slot-related phrases to overlap, to some level, with word clusters that are generated by our method.

[1] Downloaded from http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_data/muc34.tar.gz

For accomplishing the above hypothesis, we could consider a straightforward application of a clustering method to data consisting of the whole corpus vocabulary. The problem in this strategy is that many terms, which a template would be expected to identify, occur rarely in the corpus (for example: names of persons that were the target of the reported terrorist actions). In a standard clustering setting, such rare terms would not tend to be clustered together with other terms that are backed by far richer co-occurrence statistics. Potentially, terms even might function differently in different articles, e.g. the same person or group can function as the target of one event and the attacker in another event. Considerable amount of information, which is crucial to the IE task might thus be lost.

To cope with this issue, we incorporate the words of each article of the analyzed corpus as a distinct data subset, or a *component*. A clustering method, such as the IB algorithm, could be applied, as well, to the disjoint union of the article-based components. This, however, would underlie another drawback: a strong bias of words from the same article to be similar to one another, due to their frequent co-occurrences with the same feature words, relatively to words from different articles. The global influence of such bias on the results is noticeable, as the following example illustrates. Table 1 shows part of a cluster of words extracted from the MUC3/4 corpus. The displayed words are restricted to the words of one specific component (i.e. one article). Some of the words – **crisis**, **solve**, **energy**, **blackout** (Table 1A) – seem to be irrelevant to the "location" theme, characterizing this particular cluster. Presenting those words in their local context reveals some co-occurring feature words (country, occur; Table 1C), which are "location-related" and might partially account for inclusion of inappropriate elements.

Table 1. The words of a specific article that are members in one word cluster focused on "location" (the numbers indicate assignment probability or "level of membership" of each word; see Section 2 below). A: low abstraction level has been used; some irelevant words (marked by an asterisk) are present. B: using higher abstraction level, only relevant words are left. C: irrelevant words are shown in context; features that might have introduced bias are underlined.

A. Low abstraction		**B. Higher abstraction**	
central	0.9930		
western	0.9875		
zone	0.9582	**central**	0.9689
city	0.8751	**western**	0.9601
country	0.8548	**region**	0.8363
crisis*	0.8520	**zone**	0.8311
solve*	0.7922	**city**	0.7086
energy*	0.7019	**country**	0.6827
blackout*	0.6810	**san miguel**	0.6424
san miguel	0.6694		
region	0.6627		

C. The irrelevant words in context

(1) ... solve***** the country's energy***** crisis*****.

(2) ... intermittent blackouts***** have occurred in the rest of the country.

The solution to this drawback, forming the novel aspect within our method, is essentially as follows. Whenever a decision to assign a word x into a cluster c is being taken, we neglect (in part or in whole) the influence of the members of c that are also in the same component as x. Thus, the assignment of x into a cluster relies on its similarity to words occurring everywhere in the corpus except its own article. We call this bias, which we introduce into the original IB clustering method, *abstraction*, since its "abstracts" the element x being assigned from its actual context. Raising the abstraction level, our method has been able to limit the scope of the "location" cluster to relevant words (Table 1B).

In general terms, CC clustering distributes the elements (extracted words, throughout the current work) of several given components into corresponding clusters, so that the local part of each cluster within a specific component is matched with its counterparts of all other components. Each one of the resulting cross-component chains of matched cluster parts forms jointly a *cross-component cluster*. A CC-cluster would consist of elements that are similar to one another and are distinct from elements in other CC-clusters, subject to the context imposed by aligning the clustered components with respect to each other. CC clustering is intended to reflect mutual patterns that cut across the given components.

An earlier version of our method, *coupled clustering*, was designed to reveal shared structural components in a setting restricted to two data sets (Marx et al., 2002). In the current version, coupling several data sets at a time, the "hard" clustering is replaced by probabilistic "soft" assignments, which might be more appropriate for treating ambiguities such as those occurring in language. The former method takes similarity values as input with the exclusion of within-data-set similarities, which considered as representing internal regularities. The current method directly processes raw co-occurrence data. Accordingly, abstraction over the components is articulated through a more elaborated mathematical realization, which can be applied in intermediate levels, as well.

This article proceeds as follows: First, the algorithmic framework is presented (Section 2): we review the IB clustering algorithm and introduce our method – *Cross Component* (CC) clustering. Next, we describe how our method has been applied to the MUC3/4 data and our conjecture – regarding match between the output clusters and words that are assigned to MUC3/4 template slots – is verified, through examining the performance of CC clustering versus variants of the original IB method (Section 3). The paper concludes with discussing our approach in view of related work (Section 4).

2. Algorithmic Framework

Let X and Y denote respectively the set of all clustered elements and the set of all features, both consisting, throughout this work, of words extracted from the MUC3/4 corpus.

The input to the clustering algorithms – the original IB algorithm, as well as the new CC algorithm – consists of (typically sparse) feature vectors, each characterizing an element $x \in X$. A vector entry, corresponding to a feature $y \in Y$, is a conditional probability of the form $p(y|x)$. The value of each $p(y|x)$ is straightforwardly estimated based on the element-feature co-occurrence counts extracted from the examined corpora (maximum likelihood estimation):

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

The IB and CC algorithms perform soft (probabilistic) clustering. Let C denote the set of clusters. The size of C, i.e. the number of clusters, is specified as an additional input parameter. Both algorithms assign probabilistically each element $x \in X$ into a cluster $c \in C$ with probability $p(c|x)$, such that, for every x, $\sum_c p(c|x) = 1$. The collection of all $p(c|x)$ values forms the output of both algorithms.

2.1 The Information Bottleneck Algorithm

The IB clustering method is a recent approach to soft (probabilistic) clustering, in the conventional setting of a single unified set of elements (Tishby, Pereira & Bialek, 1999). The IB algorithm receives as input the set X (and a target number of clusters). Each element $x \in X$ is characterized by a probabilistic feature vector, with an entry of the form $p(y|x)$ for every feature y.

The IB method is motivated by information theoretic considerations. The algorithm minimizes a cost function:

$$\mathcal{L} = I(C;X) - \beta I(Y;C),$$

where $I(\cdot,\cdot)$ denotes mutual information of two variables (Cover & Thomas, 1991), and β is a formal Lagrange multiplier. Minimizing this weighted combination of mutual information terms implies that the algorithm seeks an optimal configuration of $p(c|x)$ values which balances the following two factors:

(i) The features provide maximal information regarding the clustering configuration, as expressed by $I(Y;C)$. That is, in a preferred configuration each cluster is characterized by a distinct set of features.

(ii) The elements provide minimal information regarding the clustering configuration, as expressed by $I(C;X)$. Intuitively, this factor implies a tendency towards spreading the elements among the clusters, as evenly as possible given the first factor (a maximum-entropy-like criterion).

The tradeoff between the two opposing factors is mediated through the multiplier β (which can be interpreted as an "inverse computational temperature" as explained in more detail below).

The IB algorithm starts by setting, for time step $t = 0$, random $p(c|x)$ values (or, alternatively, initializing them according to some heuristic). Then, it iterates, in an EM-like process (Expectation-Maximization; Dempster, Laird & Rubin, 1977), the following steps until convergence, which is guaranteed for a local minimum:

IB1. For each cluster c, compute 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 $p(y|c)$:

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

($p(x|c)$ is computed through Bayes' rule).

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

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

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

The value calculated by IB3 is a normalized product of two values: (i) the cluster prior probability $p(c)$; and (ii) the degree of similarity between the probabilistic feature vector of the element x and the vector of characteristic features of the cluster c, i.e. the $p(y|c)$ values for all $y \in Y$.

Note that the "inverse temperature" parameter β controls the sensitivity of the clustering procedure to differences between the $p(y|c)$ values. As β increases, the assignment of elements to clusters (through $p(c|x)$) becomes more sensitive to the $p(y|c)$ values, representing higher confidence in the current cluster's characteristic features. The higher β is, the more "determined" the algorithm becomes in assigning an element x into its most appropriate cluster, according to its vector of probabilities $p(y|x)$. Accordingly, when β is increased, convergence of the algorithm yields a greater number of clusters that are separable from each other (among the fixed number of all clusters). The IB algorithm is hence applied repeatedly, in a cooling-like process: it starts with a low β value, which is increased every repetition of the whole iterative converging cycle, till the desired number of separate clusters is obtained (Tishby, Pereira & Bialek, 1999).

2.2 The Cross-component Clustering Algorithm

The CC clustering algorithm receives as input a dataset X, which is the disjoint union of several components $X = \bigcup_i X_i$ (and a target number of clusters). The data elements are characterized by feature vectors, as defined at the top of Section 2. It produces the same output as the soft IB clustering algorithm, namely probabilistic assignments values $p(c|x)$ of the data elements. The output clusters are meant to accomplish, as much as possible, the task of revealing themes that are prominent across the components.

As mentioned before, the original IB algorithm can be utilized unaltered to multiple-component setting, simply by applying it to the unified dataset X, while ignoring component boundaries. In order to inspect the relative part of any component within a cluster the resulting clusters can then be projected on this component. The problem with this simplistic approach is that each component has its own characteristic features, which might, for instance, reflect the main topic discussed in the corresponding article, in contrast to the themes that cut across the whole corpus. Thus the IB method, or any other standard clustering method, would have a strong tendency to cluster together elements that originate in the same component, producing clusters that are populated mostly by whole components corresponding to articles that discuss, in whole, similar topics (cf. Marx et al, 2002).

To understand this issue in greater detail, recall that the assignment of an element x to a cluster c (step IB3) is determined by the similarity of their characterizing distributions, $p(y|x)$ and $p(y|c)$. The problem lies in the use of $p(y|c)$ to characterize a cluster, because this distribution is determined by a combination of $p(y|x)$ values over all clustered elements, *without* taking into account component boundaries. Thus, a certain y may have a high $p(y|c)$ value even though it is characteristic only for clustered elements that originate from a single "dominant" component. Since a randomly chosen element x being assigned to c with high probability typically high $p(y|x)$ values corresponding to the high $p(y|c)$ values, the overly dominant component is likely (on average) to be the component to which x itself belongs. As a result, the IB algorithm is likely to favor clusters that focus on whole components, as discussed above.

The goal of cross-component clustering is to neutralize this tendency and to create "balanced" clusters that share common features *across* different components. To overcome the unwanted tendency, it is necessary to change the criterion by which elements are assigned into clusters. For this we defined a biased probability distribution, $\tilde{p}^i(y|c)$, over all features y, defined separately for every component X_i and given a cluster c. It is used by the CC clustering algorithm to characterize a cluster c, whenever the assignment of elements of X_i is considered. $\tilde{p}^i(y|c)$ is defined such that its value for a certain y captures the degree to which $p(y|x)$ values are high, on average, across the different datasets represented within the cluster, expect the i-th component X_i. It thus tends to have high values only for y's that are typical for cluster members across different components. Consequently, an element x would be assigned into a cluster c (in an assignment step equivalent to IB3) in accordance to the degree of similarity between its own characteristic features and those that are characteristic for the cluster members originating in *different* components. The resulting clusters would tend to contain elements from all components, sharing relatively high proportion of common features.

The actual definition of $\tilde{p}^i(y|c)$ can be understood as a result of repeated updating iterations that gradually eliminate the specific unwanted part, namely $p(y|c,X_i)$, from $p(y|c)$. (We provide the details of computing $p(y|c,X_i)$ and another required value – $p(X_i)$ – in an Appendix below). We first set a temporary variable $p' = p(y,c)$, and then modify it by iterating the following step, with $k = 1,2,\ldots$:

$$p' \leftarrow p' - p(c,X_i)^k p(y|c,X_i) + p(c,X_i)^k p(y|c) .$$

Subtracting the unwanted X_i-related part still leaves p' with a positive value. However, if we wish to maintain the marginal probability $p(c)$ over all y-s, we need to add back something of the original distribution, proportioned to occupy exactly the subtracted volume (the right hand side summand). This compensation reintroduces a portion of the component that we wish to neutralize, multiplied by $p(c,X_i)^{k+1}$. Consequently, the above updating step should reiterate ad infinitum. Summing up the repeated adjustments to a geometric series, with quotient equal to $p(c,X_i)$, we now define the following synthetic joint distribution:

$$\tilde{p}^i(y,c) = p(y,c) - \frac{p(c,X_i)}{1-p(c,X_i)}\big(p(y \mid c,X_i) - p(y \mid c)\big).$$

From here, we re-normalize by $p(c)$ to obtain a conditional probability, which would be used in our analogue to step IB3:

$$\tilde{p}^i(y|c) = \tilde{p}^i(y,c) / p(c) .$$

For a relaxed version of the same type of adjustment, we similarly define, for any $0 \le \gamma \le 1$:

$$\tilde{p}^{i,\gamma}(y,c) = p(y,c) - \gamma\frac{p(c,X_i)}{1-p(c,X_i)}\big(p(y \mid c,X_i) - p(y \mid c)\big),$$

and correspondingly:

$$\tilde{p}^{i,\gamma}(y|c) = \tilde{p}^{i,\gamma}(y,c) / p(c) .$$

We term the parameter γ *abstraction level* since it represents gradual abstraction of each element from its concrete context within its component.

Having a definition for $\tilde{p}^i(y|c)$, we can present the CC clustering algorithm, which performs three iterative steps corresponding to the IB steps:

CC1. For each cluster c, compute marginal probability (same as in step IB1 of the IB algorithm):

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

CC2. Compute $\tilde{p}_t^{\,i}(y|c)$ (or $\tilde{p}^{\,i,\gamma}$) as described above.

CC3. Compute $p_{t+1}(c|x)$, with $\tilde{p}_t^{\,i(x)}(y|c)$ (or $\tilde{p}^{\,i(x),\gamma}$) playing the role played by $p(y|c)$ in step IB3 of the IB algorithm:

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

where $SIM_t^{Y,\beta}(x,c) = \exp\{-\beta\, D_{KL}[\, p(y|x) \,\|\, \tilde{p}_t^{\,i(x)}(y|c)\,]\}$ and $i(x)$ denotes the index of the component to which the element x belongs.

The same cooling process described in the context of the IB algorithm, i.e repeated runs with gradually increased β values till the desired number of separable clusters is obtained, has been applied for the CC algorithm as well.

We do not present a proof of convergence. Our experimentation with synthetic and real-world data demonstrates, however, convergence for any $0 \le \gamma \le 1$ (but not if γ is slightly greater than 1, so that positive $\tilde{p}^{\,i,\gamma}(y|c)$ values are not guaranteed).

3. CC clustering for Template Discovery

This section exemplifies how the CC clustering algorithm described above is applied to real world data, specifically for discovering IE template slots pertaining to a given (arbitrary) domain, based on an corpus of unannotated relevant articles.

Prior to testing our method on data extracted from the MUC3/4 corpus, we have applied some pre-processing to the corpus articles: lemmatizing and part-of-speech tagging (using TreeTagger[2]); removal of few function words; identification of some short word sequences as composite terms (e.g. attaching numeral values to successive nouns and sequences of terms that are labeled by TreeTagger 'NP', which correspond in many cases to proper names).

3.1 The Clustered Elements and the Features

The set of clustered elements, X, consists of all identified composite terms, proper names, nouns, verbs and adjectives extracted from the corpus articles, with little exclusion, e.g. of some frequent verbs and adjectives. The set X is a *disjoint* union of N components, corresponding to the corpus articles. In order to keep all components disjoint, we consider any word appearing in two or more of the components, as a distinct element within each component. This representation enables to capture different connotations that a certain word might have in distinct articles. Specifically, our experiments have been conducted on 217 of the 1300 MUC3/4 articles. We have chosen the articles of 200 words or more, which also fit well into the given template (specifically, 14 or more of the template slots marked as relevant in the training files with regard to each one of these articles). Thus, we have $N = 217$ components, each of which of 200-800 elements.

Each element x is characterized by a vector of features, i.e. words with which it co-occurs in the text. The features, in distinction from the elements, form one unified set Y that is shared by *all* data, thus providing common grounds for comparing elements of distinct components. The feature set Y consists of words that appear in at least three distinct articles of the processed 217. In most cases, the features incorporate their position in the text, relatively to elements with which they co-occur (e.g. in/Before vs. in/After). Some of the features also include a part of speech tag. It is hence possible to differentiate distinct prevalent senses of the same word (e.g. right/Noun vs. right/Adjective). For any particular occurrence of a clustered element x in the text, the first preceding and succeeding nouns and verbs in the same sentence are counted as features, provided they are members in Y, no matter how far apart they are from x. Features that are function words and adjectives are counted only if they are immediate neighbors of the element with which they co-occur.

Since the clustered components are disjoint, an element that appears once in an article would be represented within the corresponding component by a co-occurrence vector of *at most* 6 non-zero entries. To compensate for this intrinsic sparseness, we have added "self containment" feature (of reduced weight) for each one of the 3000 most frequent clustered terms, shared by all instances of the term in the different articles, and also by composite terms containing it (e.g. ecuadorian/Contained is a feature of ecuadorian_capital).

Modification of the details of extracting the cc-clustering input, as described above, might significantly affect the results and should be studied within subsequent work. The actual selection of particular articles, data elements and features was primarily guided by our intuition regarding the nature of the task to be accomplished, with partial support from some preliminary experimentation. We have chosen, however, to investigate more deeply two slightly diverged feature sets, which we have expected to direct interestingly altered results. The difference between the two examined features sets lies in the counting scheme of content word, specifically nouns and verbs, as features. In the first scheme – *right/left content-word context* – we count occurrences of the same verb or noun to the left and to the right of an element as different features (e.g. kill/Before vs. kill/After). Intuitively, this scheme seems to us aligned with the IE task, which requires rather detailed information on every term. The second scheme –

[2] TreeTagger – a language independent part-of-speech tagger – available for download from http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html.

unified content-word context – does not differentiate left and right occurrences of nouns or verbs. Left and right function-word features have been differentiated by both schemes. Using the unified scheme, a bit richer co-occurrence statistics is obtained.

We have applied the CC clustering algorithm, to the co-occurrence data extracted from the MUC3/4 corpus using the two co-occurrence schemes described above. We have used abstraction level, γ, ranging from 0 – which is identical to the original IB method applied to the disjoint union of components – through 0.1, 0.2, etc., up to 1. We have also applied the original IB method to a single set generated by the *non-disjoint* union of all components, using the co-occurrence counts of each element accumulated over all 217 articles. We have tested each one of the above variants with producing configurations of 20, 25, 30 and 35 clusters.

3.2 Evaluation Measures

The MUC3/4 template includes 24 slots, of which we have utilized only those involving phrases that the training files quote word-by-word from the source articles. We have extracted these phrases from the files to form reference term sets, to be compared to the output of our algorithm. The semantic themes reflected by these slot-related phrases are not obliged to encompass *all* sub-topics that might be identified through a clustering mechanism. Moreover, the slots might include subtleties that cannot be captured through clustering. For example: nearly identical phrases are used for the two slots pertaining at "identity of persons affected by the terrorist event" and "number of affected persons". We have unified the sets of phrases associated with such closely related slots, since our clusters could not distinguish the phrases associated with them, if considered separately. Column A of Table 2 assigns a title to each of the resulting phrase sets and details the ID number of original template slots to which each phrase set corresponds. Column B exemplifies some of the (shortest) phrases forming these sets. In total, we have extracted seven separate phrase sets to which we refer hereinafter as "the slots" although some of them consist of phrases associated with two or three MUC3/4 template slots.

Table 2. A sample from the slot-related phrase sets extracted from the MUC 3/4 corpus and their associated CC clusters. Numbers attached to each title indicate slot ID numbers in the original MUC template. Exemplifying items from clusters associated with each slot are on the right hand side. Items marked with an asterisk did not match the slot phrases, although some of them seem to be related to the slot topic.

A. The slot titles	B. Selection of slot phrases	C. Items from associated clusters
Location (#3)	ecuador: quito (city); itagui (municipality); sevilla (municipality)	**ecuadoran capital***; **itagui**; **370 km***
Instrument (#6-7)	truck; machetes, axes; 5,000 kg of dynamite;	**activate***; **5,000 kg**
Individual (#9)	well-dressed young men; callers; pablo escobar gaviria	**young**; **identification***; **16-year***
Organization (#10-11)	shining path; nationalist republican alliance; the government;	**shining**; **path**; **group***; **alliance**; **tupac amaru***
Physical target (#12-13-16)	electricity pylons; guard post; francisco merino's residence	**electricity**; **post**; **highway***
Human target (#18-20-23)	todd ray wilson; joaquin lopez; 15-year-old daughter	**todd**; **joaquin lopez**; **15-year**
Human target description (#19*)	leftist presidential candidate; party leader; jesuit priest	**leftist**; **leader**; **christian***

As expected, some of the word clusters produced within our experiments seem to be in good accordance with the reference slot phrase sets described above. In what follows, we give the details of how we have quantified the level of this accordance.

Our method outputs assignment probabilities ($p(c|x)$ values) rather than definite membership indication. For purposes of quantitative evaluation, we set, however, clear bounds for each cluster. One alternative for doing this is to set a threshold T, such that $p(c|x) > T$ would imply counting x as a member in c. We have found that applying a high abstraction level typically results in relatively small clusters (low $p(c)$), supplemented by an additional large cluster, which is not focused on any particular topic. Consequently, applying such a threshold would provide our method with a discriminatory advantage: small clusters tend to be cohesive and accurate, or in other words, to achieve large proportion of 'hits'. Indeed, pursuing this alternative, our method definitely outperforms the versions with low or no abstraction. Therefore, for appropriate evaluation, we have taken the 40 elements of highest $p(c|x)$ scores (no matter from which component) to be regarded as the members of c. In most cases, the lowest $p(c|x)$ values among the first 40 elements are found not very small ($p(c|x) > 0.005$).

The level of cluster-slot association is quantified by *hit proportion*, which is defined to be:

> *The proportion of cluster terms that are contained, possibly as sub-phrases, within the slot-related phrases that have been extracted from each term's source article.*

It turns that for most slots, there is a best-matched cluster, of relatively high hit proportion, which is, in most cases, unique and well differentiated. I.e., a typical best-matched cluster is noticeably superior to the second-best cluster and it is not a best match for any additional slot. Hit proportions of best-matched clusters might range from as little as 5% of the cluster members to 40% and more. The best-matched cluster typically contains additional terms that are related to its slot's theme, but are not contained in the filled template phrases (examples for such

terms are marked by asterisks in table 2C). There are additional interesting factors that are not covered through hit proportion, such as the quality of clusters that are not matched with any slot. Some of the clusters reveal themes that could have been, but are not, reflected by the template (for example: a cluster consisting of terms that are related with political processes: **dialogue**, **solution**, **negotiation**, **democratization** and so on). However, evaluating the quality of non-matched clusters or the exact relevance of each term is a subjective, non-automated task. On the other hand, it is apparent that a typical best-matched cluster differentiates its slot from the other ones. Hit proportion, partial and rough as it is, thus provides a clear and objective indication of how well output clusters by the various algorithms capture topics that are associated with the slots.

We explicate two distinct overall quality measures, both are based on the hit proportion defined above. The first one is a global *covering* measure. Specifically, the overall level by which the output clusters capture the various slot-related themes is the hit proportion of a slot's best-matched cluster, averaged over all seven slots.

In addition, we suggest a measure of *differentiation*, indicating how well the output clusters configuration identifies each slot-related theme as a distinguishable theme. Considering one particular slot S, we would like to quantify the distribution of S-related terms among all clusters. The entropy of the normalized distribution of hit proportions corresponding to S, over all clusters, measures the differentiability of particular theme reflected by S. It gives an indication of how the phrases related to one S are concentrated within one or few clusters: the lower the entropy the better is the differentiation. (Illustratively, suppose that S has a best-matched cluster with hit proportion of 20%, but there are additional clusters of hit proportions 2%, 5%, 10% and 15% with S, which must be considered before concluding whether S was "detected" by the given clustering configuration). The entropy is formally defined as follows:

$$\mathrm{Ent}(S) = -\sum_c \rho_S(c) \log\left(\rho_S(c)\right),$$

where $\rho_S(c) = HP(c,S) / \sum_{c'} HP(c',S)$ and $HP(c,S)$ is the hit proportion of the cluster c with regard to the slot S.

We can average Ent(S) over the seven slots, to obtain a global measure. A problem, however, arises when we want to compare averaged Ent values across configurations of different numbers of clusters. The solution lies in using a different, but closely related, measure:

$$\mathrm{DfUD}(S) = \log(K) - \mathrm{Ent}(S),$$

where K is the number of clusters and Ent is as above.

DfUD (Divergence from Uniform Distribution) is the KL divergence of the normalized distribution $\rho_S(c)$ from a uniform distribution over the K clusters. Its sign is, of course, opposite to Ent: the higher a DfUD value is, the better differentiation obtained. It produces values that are comparable over configurations of different cluster numbers, so we take its average over all slots to be a global measure comparable across configurations.

3.3 Summary of Experimental Results

By means of the two measures – *best hit-proportion* and *divergence from uniform distribution* – we compare the varying factors that we have tested: changing abstraction levels, number of clusters and co-occurrence counting scheme. The patterns of response to changes in the various parameters, relatively to any particular slot, show occasional fluctuations. The global measures averaged over the seven slots still result in a noisy depiction (Figure 1). Some preliminary observations could nevertheless be drawn.

The most notable finding is that clustering of a unified data set with accumulative co-occurrence counts (left hand side of each plot in Figure 1) is inferior to most examined variants of multi-set approach, despite the significantly richer statistics characterizing frequent terms in the unified setting. In particular, the unified-set clusters tend to be characterized (i.e. to have high $p(y|c)$ values) by features that are relatively prevalent, such as have, say, in, and so on. In contrast, the multi-component settings seem, in general, to capture better the context of infrequent terms and names, which play significant role in IE and, in particular, in the MUC-3/4 data.

Relatively good average hit-proportion results are obtained in the co-occurrence count scheme, distinguishing left from right content-word occurrences. These results deteriorate, however, starting at abstraction levels of 0.5-0.6 and up, particularly in the cases of fewer clusters (Figure 1A). At the same time, the averaged divergence from uniform distribution shows average constant improvement as abstraction level increases (Figure 1B). A conceivable explanation is that the data that underlies formation of clusters is so sparse in this setting to the extent that the clusters lose many of their relevant elements, even among their first 40 members. On the other hand, the reduced core of each cluster preserves its discriminative power and even enhances it.

In the other co-occurrence count scheme, which does not distinguish left and right occurrences of content words, best hit-proportion maintains its level while abstraction level increases (Figure 1C). On the other hand, improvement in divergence from uniform distribution measure is slight (Figure 1D), and in general inferior to that of the other count scheme.

Based on the above findings, we hence suggest tentative directions of using the differentiating co-occurrence count mode, which we have considered better tuned to the task in the first place, with middle range abstraction level, such as 0.5, to preserve both hit-level and discriminative power. More experiments, however, are needed in order to put these directions on more definite grounds.

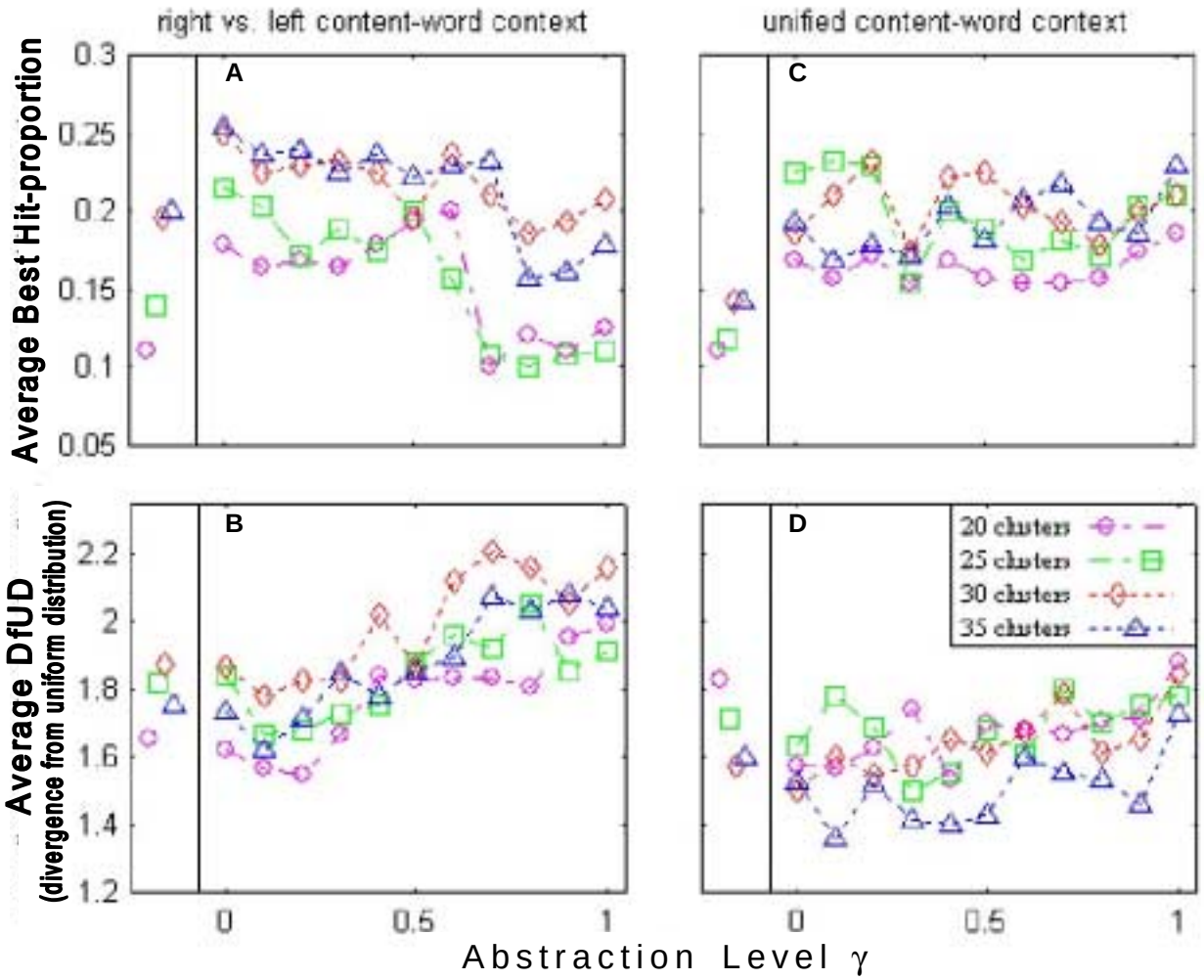

Figure 1. Average results for configurations of 20, 25, 30 and 35 clusters. The top plots show best cluster hit proportion. The bottom plots show the average divergence, relatively to each slot, of the hit proportion distribution from uniform distribution. The left and right hand side plots differ by the counting method of verb and noun feature co-occurrence: different left and right counts vs. unified count respectively. The left-hand side of each plot displays results from clustering of a single unified set.

4. Related Work and Discussion

The approach presented in this paper still requires considerable amount of additional work. For instance, further study may reveal whether abstraction can be applied to any co-occurrence based clustering method, similarly to the modification of the IB algorithm illustrated here. We have found the IB algorithm attractive, because of its principled interpretation within information theory, which can potentially be found relevant also for CC clustering. The IB approach has already been extended in several ways, which might appear related to the CC clustering method. The *multivariate information bottleneck* setting (Friedman et al., 2001) handles, similarly to our method, elements of several sets. The difference is that the multivariate setting assumes simultaneous multi-occurrences of representatives from all feature and data sets, while our method takes co-occurrences of one feature with one element of any of the components at a time. The multivariate setting allows also simultaneous clustering of the data in several clustering configurations, according to several criteria. Combining it with CC clustering could turn a fruitful direction.

The idea of differentiating the various contexts of each clustered word, rather than applying a clustering procedure to the unified set of all words in corpus, is not entirely new. Schütze (1998) has applied a clustering algorithm to the set of co-occurrence vectors representing the specific occurrences of a single word, for the purpose of *word sense disambiguation*, which is restrictedly related with our IE task. For example, the word *car* could be regarded ambiguous in the sense that it can be assigned to the "instrument" slot in one article and to the "physical target" slot in another article. Differently then Schütze, we accumulate the statistics for each word throughout

each article, where it is rather probable that it carries the same meaning.

In this paper, our method has been applied to a large collection of presumably homogenous data sets, each one of which providing rather sparse statistics. Thus we have demonstrated capabilities of CC clustering in capturing subtleties and purifying the context relevant to a specific practical task: template induction for IE. It is a matter of empirical examination to check whether CC clustering would be helpful in actual template definition. It is rather obvious that it can provide some help in the absence of any domain-specific knowledge. There is no clear evidence regarding the perspective that it can attach to approachable detailed directions. The examples that we have encountered, such as the "political process" cluster, suggest, however, that in some cases informative additional points of interest might be discovered. Anyway, the comparative results suggested in the previous section give a fairly clear indication that CC clustering has some advantage on standard clustering methods used for the same task.

To the best of our knowledge, previous research concerning unsupervised template discovery for IE has not been very intensive. Riloff (1996) examined word combinations of predefined syntactic relations and analyzed the differences of their frequency within relevant versus non-relevant texts, for identifying sub-topical key combinations – *extraction patterns*. Yangarber et al. (2000) used a similar approach. Starting from a seed of few word patterns that are known to be good indicators of the topic of interest, they avoided the need to tag in advance documents as relevant or irrelevant. Collier (1998) located words that are statistically significant within a corpus and then identified the sentences in which they occur as key sentences. Whenever the extracted sentences contained corpus-significant verbs, these sentences have been assumed corresponding to the required template, whether or not known as such to the user.

In distinction from the works mentioned above, our method is designed for a very general setting. It thus discards information that seems helpful in template induction, such as syntactic combinations or complete sentences. Further research is required to examine if and how to adapt our notion of abstraction to such informative constructs. For example: a clustering method can be applied to Riloff's (1996) extraction patterns, based on their co-occurrences with neighboring words or other constructs, aiming at homogeneous slot-related groups of patterns. Abstracting each particular instance of an extraction pattern from its within-article co-occurrence information might turn beneficial in this task, as in our case.

We expect our method to be found fruitful for additional text learning tasks. For instance, CC clustering of smaller number of components that substantially differ from one another, can be used for identification of unexpected similar aspects and creative analogies, as have been demonstrated in the earlier work on coupled clustering (Marx et al., 2002). In particular, illuminating results of applying CC clustering to the religion-related corpus used by Marx et al. would be published elsewhere.

Taking a broader perspective, we observe that the ongoing progress in machine learning methods reflects a growing attention directed towards data exploration. In recent years, the amount and complication of information deserving scientific investigation, for instance, is augmented to a level restricting the traditional way of formulation and examination of hypotheses. Complementarily, there is a growing interest in understanding and mechanizing the process of scientific discovery to allow the automated induction of hypotheses and models. In this context, we draw attention to the correspondence between data clustering and statistical inference methods, specifically *analysis of variance* (ANOVA, Scheffe, 1959). While one-way ANOVA directs inference through pre-given cells, standard clustering algorithms, e.g. IB, partitions the given data into homogenous cells, with the highest obtainable between-cell variance. In a like manner, our method forms an unsupervised analog to two-way ANOVA. Similarly to standard clustering, CC clustering reveals, through partitioning of the data, a factor that is not given in advance. Another factor – namely the prior partition into distinct components – is pre-given. The output configuration is expected to neutralize the potential interaction between the two factors. Often, such interaction is interesting and requires investigation. By way of contrast, CC clustering deals with cases were the motivation to neutralize both individual characteristics and interaction is apparent or assumed. The IE task indeed exemplifies a discipline that tries to identify common characteristics among many exemplars. Our approach thus naturally follows the view promoted along these lines, to extend IE in an unsupervised manner.

It is known that there are cases in which discarding some details of the available information would yield better results. This familiar fact is practiced ordinarily through various strategies such as smoothing. The CC clustering method can be also interpreted as a smoothing technique explicitly directed towards a particular factor that one would wish to neutralize. Data clustering methods form a helpful data exploration tool, providing that the determining features are known to be relevant to the task under study. Our method presents a complimentary framework for eliminating factors – namely within-component regularities – which are integrally embodied within the data but are known, or assumed, to be irrelevant.

Appendix

The value of $p(X_i)$, which is required for the calculations in Section 2.2, 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 value $p_t(y|c,X_i)$, which is used as well in Section 3.2, is calculated as follows from values that are available at time step $t-1$:

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

($p_{t-1}(x|c,X_i)$ is computed through 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 $p_{t-1}(c|x) = p_{t-1}(c|x,X_i)$.)

Acknowledgements

We thank Joachim Buhmann, Yuval Krymolowski and Naftali Tishby, 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

Chinchor, N., Hirschman, L., and Lewis, D. (1993). Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3), *Computational Linguistics*, *19*, 409–448.

Collier. R. (1998). *Automatic template creation for information extraction*. Doctoral dissertation, Department of Computer Science, The University of Sheffield, UK.

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

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B, 39*, 1-38.

Friedman, N., Mosenzon, O., Slonim, N. and Tishby, N. (2001). Multivariate information bottleneck. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence UAI-2001*, Seattle, WA.

Gaizauskas, R. and Wilks, Y. (1998). Information extraction: beyond document retrieval. *Journal of Documentation*, *54*, 70–105.

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. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics ACL' 93* (pp. 183–190), Columbus, OH.

Riloff., E. (1996). Automatically generating extraction patterns from untagged text. *Proceedings of Thirteenth National Conference on Artificial Intelligence AAAI-96*, (pp. 1044–1049), Portland, OR.

Schütze, H. (1998) Automatic word sense discrimination. *Computational Linguistics, 24*, 97–124.

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

Scheffe, H. (1959). *The Analysis of Variance*. New York: John Wiley & Sons.

Yangarber R., Grishman R., Tapanainen, P. and Huttunen, S. (2000). Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. *Proceeding of the Sixth Applied Natural Language Processing Conference* (pp. 282–289), Seattle, WA.