Journal of Information Assurance and Security 4 (2009) 174-182

A Text Clustering Framework for Information Retrieval

Sergio Decherchi, Paolo Gastaldo, Judith Redi and Rodolfo Zunino

Dept. Biophysical and Electronic Engineering, University of Genoa,
16145 Genoa, Italy
{sergio.decherchi, paolo.gastaldo, judith.redi, rodolfo.zunino}@unige.it

***Abstract*:** Text-mining methods have become a key feature for homeland-security technologies, as they can help explore effectively increasing masses of digital documents in the search for relevant information. This research presents a model for document clustering that arranges unstructured documents into content-based homogeneous groups. The overall paradigm is hybrid because it combines pattern-recognition grouping algorithms with semantic-driven processing. First, a semantic-based metric measures distances between documents, by combining content-based and behavioral analysis. Such a metric allows taking into account the lexical properties, the structure and the styles characterizing the processed documents. In a second step, the model relies on a Radial Basis Function (RBF) kernel-based mapping for clustering documents. As a result, the major novelty aspect of the proposed approach is to exploit the implicit mapping of RBF kernel functions to tackle the crucial task of normalizing similarities, while embedding semantic information in the whole mechanism. Experimental results on Reuters and Newsgroup 20 databases validate the proposed approach.

***Keywords*:** document clustering, homeland security, kernel k-means., documents similarity, text mining, unsupervised learning

1. Introduction

The automated surveillance of information sources is of strategic importance to effective homeland security [1],[2]. The increased availability of data-intensive heterogeneous sources provides a valuable asset for the intelligence tasks; data-mining methods have therefore become a key feature for security-related technologies [2],[3], as they can help in effectively exploring increasing masses of digital data when searching for relevant information.

Text mining techniques provide a powerful tool to deal with large amounts of unstructured text data gathered from heterogeneous multimedia sources (e.g. Optical Character Recognition, audio via speech transcription, web-crawling agents, etc., see fig.1) [4],[5]. Text mining methods can be applied successfully in the network security domain, following several approaches [5]: detection/tracking tools can be used to continuously monitor specific topics over time; document classifiers label individual files and build up models for possible subjects of interest; relations among the selected subjects can be then detected with the help of clustering tools. As a result, text mining can profitably support intelligence and security activities in identifying, tracking, extracting, classifying and discovering patterns, so that the outcomes can generate alert notifications accordingly [6],[7]. This work addresses document clustering and presents a dynamic, adaptive clustering model to arrange unstructured documents into content-based homogeneous groups. The framework implements a hybrid paradigm, which combines a content-driven similarity processing with pattern-recognition grouping algorithms. Distances between documents are worked out by a semantic-based hypermetric: the specific approach integrates a content-based with a user-behavioral analysis, as it takes into account both lexical and style-related features of the documents at hand. The core clustering strategy exploits a kernel-based version of the conventional k-means algorithm [8]; the present implementation relies on a Radial Basis Function (RBF) kernel-based mapping [9]. The advantage of using such a kernel consists in supporting normalization implicitly; normalization is a critical issue in most text-mining applications, and prevents that extensive properties of documents (such as length, lexicon, etc) may distort representation and affect performance.

Standard benchmarks for content-based document management, the Reuters database [10] and Newsgroup 20 database [11], provided the experimental domain for the proposed methodology. The research shows that the document clustering framework based on kernel k-means can generate consistent structures for information access and retrieval.

Received June 10, 2009

1554-1010 $ 03.50 Dynamic Publishers, Inc

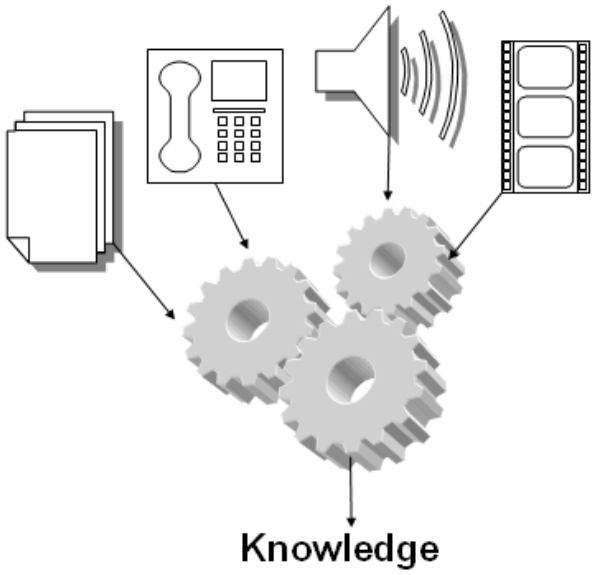

Figure 1. Different sources of data compete to produce knowledge when processed by a common clustering engine

2. Document Clustering in Text Mining

Text mining can effectively support the strategic surveillance of information sources thanks to automatic means, which is of paramount importance to homeland security [6],[7]. For prevention, text mining techniques can help identify novel "information trends" revealing new scenarios and threats to be monitored; for investigation, these technologies can help distil relevant information about known scenarios. Within the text mining framework, this work addresses document clustering, which is one of the most effective techniques to organize documents in an unsupervised manner.

2.1 Document clustering

Clustering is conventionally ascribed to the realm of pattern recognition and machine learning [12]. When applied to text mining, clustering algorithms are designed to discover groups in the set of documents such that the ones within a group are more similar to one another than to those belonging to other groups. As opposed to text categorization [5], in which predefined categories enter the learning procedure, document clustering follows an unsupervised approach to search, retrieve, and organize key topics when a proper set of categories cannot be defined *a-priori*. The unsupervised paradigm can address challenging scenarios, in which local episodes of interest can fade away in the clutter of very large datasets, where events or profiles are ambiguous, unknown, or possibly changing with respect to the original models.

The document clustering problem can be defined as follows. One should first define a set of documents $\mathcal{D} = \{D_1, ..., D_n\}$, a similarity measure (or distance metric), and a partitioning criterion, which is usually implemented by a cost function. *Flat* clustering [13] can be used to identify a set of clusters without needing any a-priori assumption about the structure among them, and it typically requires the number of clusters to be specified in advance, although methods exist for determining the cluster cardinality adaptively [14]. Once the desired number of clusters, K, is defined, a membership function $\phi : \mathcal{D} \rightarrow \{1, \ldots, K\}$ is computed such that ϕ minimizes the partitioning cost with respect to the similarities among documents. On the other hand, *hierarchical* clustering [13] arranges groups of items in a structural, multilevel fashion and does not require a pre-specified number of clusters; these advantages often come at the cost of a lower computational efficiency. Hierarchical clustering doesn't need the cardinality, K, to be defined because it applies a series of nested partitioning tasks, which eventually yield a hierarchy of groups. In addition to the selection between a flat or hierarchical partitioning strategy, three main issues should be addressed when designing the overall clustering framework.

The first issue is the dimensionality. When using a vector-space approach, documents lie in a space whose dimensionality typically ranges from several to tens of thousands. Nonetheless, most documents normally contain a very limited fraction (1%–5%) of the total number of terms included in the adopted vocabulary, hence the vectors representing documents are very sparse. This can make learning extremely difficult in such a high-dimensional space, especially due to the so-called curse of dimensionality [15]. It is typically desirable to project documents preliminarily into a lower-dimensional subspace, which preserves the semantic structure of the document space but facilitates the use of traditional clustering algorithms. Several methods for low-dimensional document projections have been proposed [16], such as spectral clustering [17], clustering using the Latent Semantic Index (LSI) [18],[19], clustering using the Locality Preserving Indexing (LPI) [20], and clustering based on nonnegative matrix factorization [21]. Those methods are quite popular but also exhibit theoretical and practical drawbacks. Both the LSI and the LPI model rely on Singular Value Decomposition (SVD) [22], which optimizes a least-square criterion and best performs when data are characterized by a normal distribution. In fact, the latter assumption does not hold in the general case of term-indexed document matrixes. Besides, LSI, LPI and spectral clustering all require the computation of eigenvalues; as such, these methods often prove both heavy from a computational viewpoint and quite sensitive to outliers. Spectral clustering has also been proved to be a special case of a weighted form of the kernel k-means [23]. Nonnegative matrix factorization (NMF) differs from other rank reduction methods for vector spaces, especially because of specific constraints that produce nonnegative basis vectors. However, the iterative update method for solving NMF problem is computationally expensive and produces a non-unique factorization.

The second issue in setting up an effective clustering process is the definition of the similarity measure. Since the partitioning criterion often relates strictly to the similarity measure, the choice of the underlying metrics is critical for getting meaningful clusters. For documents, it is normal to

address some content-based similarity, and most clustering tools adopt the vector-space model because such a framework easily supports the popular cosine similarity:

$$sim(D_i, D_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\left|\mathbf{v}_i\right|\left|\mathbf{v}_j\right|} \quad (1)$$

where $\mathbf{v}_i$ is the vector representing document D_i and the operator $(\cdot)$ denotes the conventional inner product in the vector space. The normalization implied by the denominator in (1) prevents that two documents having similar distributions of terms appear distant from each other just because one is much longer than the other. In fact, the cosine similarity seems to not outperform the conventional Euclidean distance when high dimensional spaces are concerned [24].

The third issue in clustering for text mining concerns the specific algorithm to be implemented. Although the literature offers a wide variety of clustering algorithms, the majority of research in text mining involves three approaches, namely, *k-means* clustering, Self Organizing Maps (SOMs), and the Expectation-Maximization (EM) algorithm [25], [26], [27], [28]. Alternative approaches include models based on fuzzy clustering techniques [29],[30]. Furthermore, on a slightly different perspective, the works of Hammouda et al. [31] and Chim et al.[32] proposed document-clustering schemes exploiting a phrase-based document similarity. The former scheme [31] exploits the Document Index Graph (DIG), which indexes the documents while maintaining the sentence structure in the original documents; the latter scheme [32] is based on the Suffix Tree Document (STD) model [33].

2.2 The information extraction model

Every text mining framework should always be supported by an information extraction (IE) model [13],[25], designed to pre-process digital text documents and to organize the information according to a given structure that can be directly interpreted by a machine learning system.

IE models reduce a document D to a sequence of terms and eventually represent it as a vector lying in a space spanned by the dictionary (or vocabulary) $\mathcal{T} = \{t_j; j = 1,.., n_T\}$.

The dictionary collects all terms used to represent any document D, and can be assembled empirically by gathering those words that occur at least once in a document collection $\mathcal{D}$. By this representation, the original relative ordering of terms within each document is lost. Different models [13],[25] can be used to retrieve index terms and to generate the vector that represents a document D. However, the vector space model [34] is the most widely used method for information extraction in document clustering. Given a collection of documents $\mathcal{D}$, the vector space model represents each document D as a vector of real-valued weight terms $\mathbf{v} = \{w_j; j=1,..,n_T\}$. Each component of the n_T-dimensional vector is a non-negative term weight, w_j, characterizing the j-th term and denoting the relevance of the term itself within the document D.

3. Hybrid Distance and Clustering

The hybrid approach described in this Section combines the specific advantages of content-driven processing with the effectiveness of an established pattern-recognition grouping algorithm. Document similarity is defined by a content-based distance, which combines a classical distribution-based measure with a behavioral analysis of the style features of the compared documents. The core engine relies on a kernel-based version of the classical k-means partitioning algorithm [8] and groups similar documents by a top-down hierarchical process. In the kernel-based approach, every document is mapped into an infinite-dimensional Hilbert space, where only inner products among elements are meaningful and computable. In the present case the kernel-based version of k-means [35] provides a major advantage over the standard k-means formulation.

In the following, $\mathcal{D} = \{D_u; u = 1,..,n_D\}$ will denote the corpus, holding the collection of documents to be clustered. The set $\mathcal{T} = \{t_j; j = 1,.., n_T\}$ will denote the vocabulary, which is the collection of terms that occur at least one time in $\mathcal{D}$ after the pre-processing steps of each document $D \in \mathcal{D}$ (e.g., stop-words removal, stemming [13]).

3.1 Documents distance measure

A novel aspect of the method described here is the use of a document-distance that takes into account both a conventional content-based similarity metric and a behavioral similarity criterion. The latter term aims to improve the overall performance of the clustering framework by including documents structure and style information in the process of similarity evaluation. To support the proposed document distance measure, a document D is here represented by a pair of vectors, $\mathbf{v}'$ and $\mathbf{v}''$. Vector $\mathbf{v}'(D)$ addresses the content description of a document D; it can be viewed as the conventional n_T-dimensional vector that associates each term $t \in \mathcal{T}$ with the normalized frequency, tf, of that term in the document D. Therefore, the k-th element of the vector $\mathbf{v}'(D_u)$ is defined as:

$$v'_{k,u} = tf_{k,u} \Big/ \sum_{l=1}^{n_T} tf_{l,u} \quad (2)$$

where $tf_{k,u}$ is the frequency of the k-th term in document D_u. Thus, $\mathbf{v}'$ represents a document by a classical vector model, and uses term frequencies to set the weights associated to each element.

From a different perspective, the structural properties of a document, D, are represented by a set of probability distributions associated with the terms in the vocabulary. Each term $t \in \mathcal{T}$ occurring in D_u is associated with a distribution function that gives the spatial probability density function (pdf) of t in D_u. Such a distribution, $p_{t,u}(s)$,

is generated under the hypothesis that, when detecting the k-th occurrence of a term t at the normalized position $s_k \in [0,1]$ in the text, the spatial pdf of the term can be approximated by a Gaussian distribution centered around s_k. In other words, if the term t_j is found at position s_k within a document, a second document with similar structure is expected to include the same term at the same position or in a neighborhood thereof, with a probability defined by a Gaussian pdf. To derive a formal expression of the pdf, assume that the u-th document, D_u, holds n_O occurrences of terms after simplifications; if a term occurs more than once, each occurrence is counted individually when computing n_O, which can be viewed therefore as a measure of the length of the document. The spatial pdf can be defined as:

$$p_{t,u}(s) = \frac{1}{A}\sum_{k=1}^{n_O} G(s_k, \lambda) = \frac{1}{A}\sum_{k=1}^{n_O} \frac{1}{\sqrt{2\pi\lambda}} \exp\left[-\frac{(s-s_k)^2}{\lambda^2}\right] \quad (3)$$

where A and λ are normalization terms. In practical situations, one uses a discrete approximation of (3). First, the document D is segmented evenly into S sections. Then, an S-dimensional vector is generated for each term $t \in \mathcal{T}$; each element of that vector estimates the probability that the term t occurs in the corresponding section of the document (fig.2). As a result, $\mathbf{v}''(D)$ is an array of n_T vectors having dimension S (fig.3).

Vectors $\mathbf{v}'$ and $\mathbf{v}''$ support the computation of the frequency-based distance, $\Delta(f)$, and the behavioral distance, $\Delta(b)$, respectively.

The former term is usually measured according to a standard Minkowski distance, hence the content distance between a pair of documents (D_u, D_v) is defined by:

$$\Delta^{(f)}(D_u, D_v) = \left[\sum_{k=1}^{n_T} \left| v'_{k,u} - v'_{k,v} \right|^p \right]^{1/p} \quad (4)$$

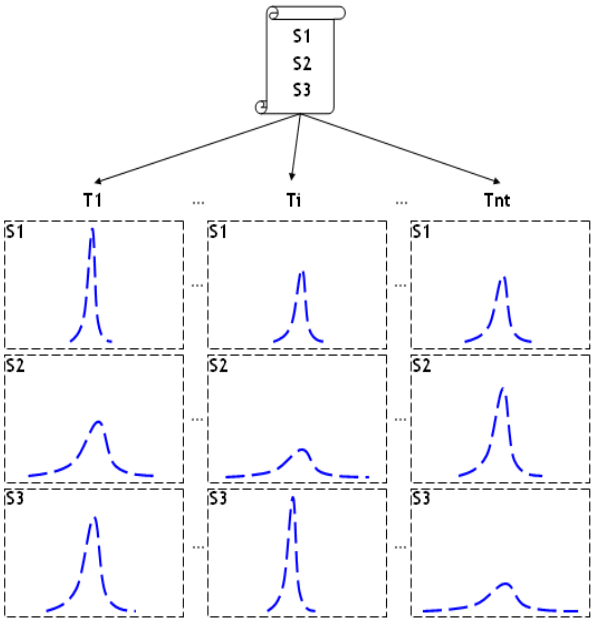

Figure 2. A document partitioned in 3 sections and terms $\mathcal{T} = \{t_j; j = 1,.., n_T\}$ Gaussian densities in each section

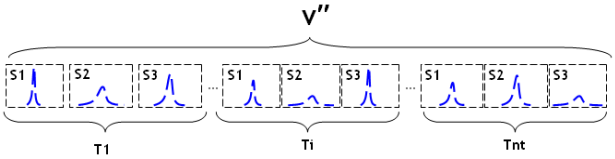

Figure 3. A document partitioned in 3 sections, terms $\mathcal{T} = \{t_j; j = 1,.., n_T\}$ and vector $\mathbf{v}''$ representation

The present approach adopts the value $p = 1$ and therefore actually implements a Manhattan distance metric. The term computing behavioral distance, $\Delta(b)$, applies an Euclidean metric to compute the distance between probability vectors $\mathbf{v}''$. Thus:

$$\Delta^{(b)}(D_u, D_v) = \sum_{k=1}^{n_T} \Delta^{(b)}_{t_k}(D_u, D_v) = \sum_{k=1}^{n_T}\sum_{s=1}^{S} \left[v''_{(k)s,u} - v''_{(k)s,v} \right]^2 \quad (5)$$

Both terms (4) and (5) contribute to the computation of the eventual distance value, $\Delta(D_u,D_v)$, which is defined as follows:

$$\Delta(D_u,D_v) = \alpha \cdot \Delta^{(f)}(D_u,D_v) + (1-\alpha) \cdot \Delta^{(b)}(D_u,D_v) \quad (6)$$

where the mixing coefficient $\alpha \in [0,1]$ weights the relative contribution of $\Delta(f)$ and $\Delta(b)$. It is worth noticing that the distance expression (6) obeys the basic properties of non-negative values and symmetry that characterize general metrics, but does not necessarily satisfy the triangular property.

3.2 Kernel k-means

The conventional k-means paradigm supports an unsupervised grouping process [8], which partitions the set

of samples, $\mathcal{D} = \{D_u;\ u = 1,..,n_D\}$, into a set of Z clusters, C_j ($j = 1,\ldots, Z$). In practice, one defines a "membership vector," which indexes the partitioning of input patterns over the K clusters as: $m_u = j \Leftrightarrow D_u \in C_j$, otherwise $m_u = 0$; $u = 1, \ldots, n_D$. It is also useful to define a "membership function" $\delta_{uj}(D_u, C_j)$, that defines the membership of the u-th document to the j-th cluster: $\delta_{uj} = 1$ if $m_u = j$, and 0 otherwise. Hence, the number of patterns belonging to a cluster is expressed as:

$$N_j = \sum_{u=1}^{n_D} \delta_{uj}\ ; \quad j = 1,\ldots, Z\ ; \tag{7}$$

and the cluster centroid is given by:

$$\mathbf{w}_j = \frac{1}{N_j} \sum_{u=1}^{n_D} \mathbf{x}_u \delta_{uj}\ ; \quad j = 1,\ldots, Z\ ; \tag{8}$$

where $\mathbf{x}_u$ is any vector-based representation of document D_u. The following cost has to be minimized :

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{Z} \sum_{\mathbf{x}_j \in \mathbf{C}} \| \mathbf{x}_j - C_i \|^2 \tag{9}$$

The kernel-based version of the algorithm relies on the assumption that a function, Φ, can map any element, D, into a corresponding position, $\Phi(D)$, in a possibly infinite dimensional Hilbert space. The mapping function defines the actual 'Kernel', which is formulated as the expression to compute the inner product:

$$K(D_u, D_v) = K_{uv} = \Phi(D_u) \cdot \Phi(D_v) \overset{def}{=} \Phi_u \cdot \Phi_v \tag{10}$$

In our particular case we employ the largely used RBF kernel:

$$K(D_u, D_v) = \exp\left[-\frac{\Delta(D_u, D_v)}{\sigma^2} \right] \tag{11}$$

It is worth stressing here an additional, crucial advantage of using a kernel-based formulation in the text-mining context: the approach (11) can effectively support the critical normalization process by reducing all inner products within a limited range, thereby preventing that extensive properties of documents (length, lexicon, etc) may distort representation and ultimately affect clustering performance. The kernel-based version of the k-means algorithm, according to the method proposed in [35], replicates the basic partitioning schema (7)-(8) in the Hilbert space, where the centroid positions, Ψ, are given by the averages of the mapping images, Φ_u:

$$\Psi_j = \frac{1}{N_j} \sum_{u=1}^{n_D} \Phi_u \delta_{uj}\ ; \quad j = 1, \ldots, Z\ . \tag{12}$$

The ultimate result of the clustering process is the membership vector, m, which determines prototype positions (8) even though they cannot be stated explicitly. As a consequence, for a document, D_u, the distance in the Hilbert space from the mapped image, Φ_u, to the cluster Ψ_j as per (8) can be worked out as:

$$\begin{aligned} & d(\Phi_u, \Psi_j) = \\ & = \left\| \Phi_u - \frac{1}{N_j} \sum_{v=1}^{n_D} \Phi_v \right\|^2 \\ & = 1 + \frac{1}{(N_j)^2} \sum_{m,v=1}^{n_D} \delta_{mj} \delta_{vj} K_{mv} - \frac{2}{N_j} \sum_{v=1}^{n_D} \delta_{vj} K_{u,v} \end{aligned} \tag{13}$$

By using expression (13), which includes only kernel computations, one can identify the closest prototype to the image of each input pattern, and assign sample memberships accordingly.

In clustering domains, k-means clustering can notably help separate groups and discover clusters that would have been difficult to identify in the base space. From this viewpoint one might even conclude that a kernel-based method might represent a viable approach to tackle the dimensionality issue.

4. Experimental Results

The experimental session involved the analysis of Reuters [10] and Newsgroup 20 [11] datasets, being well known and widely accepted benchmarks for text mining problems. Moreover, those data represent a very heterogeneous source of information and make possible to explore in a large range of problems the capabilities of the developed framework.

In the following experiments, the performances of the clustering framework have been evaluated by using the purity parameter. Let N_k denote the number of elements lying in a cluster C_k and let N_{mk} be the number of elements of the class I_m in the cluster C_k. Then, the purity $pur(k)$ of the cluster C_k is defined as follows:

$$pur(k) = \frac{1}{N_k} \max_m (N_{mk}) \tag{14}$$

Accordingly, the overall purity of the clustering results is defined as follows:

$$purity = \sum_k \frac{N_k}{N} \cdot pur(k) \tag{15}$$

where N is the total number of element. The purity parameter has been preferred to other measures of performance (e.g. the F-measures) since it is the most accepted measure for machine learning classification problems [11].

4.1 Reuters dataset

A standard benchmark for content-based document management, the Reuters database [10], provided one of the experimental domains for the proposed framework. The database includes 21,578 documents, which appeared on the Reuters newswire in 1987. One or more topics derived from economic subject categories have been associated by human indexing to each document; eventually, 135 different topics were used. In this work, the experimental session involved a corpus $\mathcal{D}_R$ including 8267 documents out of the 21,578 originally provided by the database. The corpus $\mathcal{D}_R$ was obtained by adopting the criterion used in [36]. First, all the documents with multiple topics were discarded. Then, only the documents associated to topics having at least 18 occurrences were included in $\mathcal{D}_R$. As a result, 32 topics were represented in the corpus.

The clustering performance of the proposed methodology was evaluated by analyzing the result obtained with three different experiments: the documents in the corpus $\mathcal{D}_R$ were partitioned by using a flat clustering paradigm and three different settings for the parameter α, which, as per (6), weights the relative contribution of $\Delta(f)$ and $\Delta(b)$ in the document distance measure. The values used in the experiments were α = 0.3, α = 0.7 and α = 0.5; thus, a couple of experiments were characterized by a strong preponderance of one of the two components, while in the third experiment $\Delta(f)$ and $\Delta(b)$ evenly contribute to the eventual distance measure.

Table 1 outlines the results obtained with the setting α = 0.3. The evaluations were conducted with different number of clusters *Z*, ranging from 20 to 100. For each experiment, four quality parameters are presented:

- the overall purity, $purity_{OV}$, of the clustering result;
- the lowest purity value $pur(k)$ over the *Z* clusters;
- the highest purity value $pur(k)$ over the *Z* clusters;
- the number of elements (i.e. documents) associated to the smallest cluster.

Analogously, Tables 2 and 3 reports the results obtained with $\alpha = 0.5$ and $\alpha = 0.7$, respectively.

As expected, the overall purity grows with number of clusters *Z*. Indeed, the value of the overall purity seems to indicate that clustering performances improve by using the setting $\alpha = 0.3$. Hence, empirical outcomes confirm the effectiveness of the proposed document distance measure, which combines the conventional content-based similarity with the behavioral similarity criterion.

Number of clusters	Overall purity	$pur(k)$ minimum	$pur(k)$ maximum	Smallest cluster
20	0.712108	0.252049	1	109
40	0.77138	0.236264	1	59
60	0.81154	0.175	1	13
80	0.799685	0.181818	1	2
100	0.82666	0.153846	1	1

Table 1. Clustering performances obtained on Reuters-21578 with α=0.3.

Number of clusters	Overall purity	$pur(k)$ minimum	$pur(k)$ maximum	Smallest cluster
20	0.696383	0.148148	1	59
40	0.782267	0.222467	1	4
60	0.809121	0.181818	1	1
80	0.817467	0.158333	1	1
100	0.817467	0.139241	1	2

Table 2. Clustering performances obtained on Reuters-21578 with α=0.5.

Number of clusters	Overall purity	$pur(k)$ minimum	$pur(k)$ maximum	Smallest cluster
20	0.690577	0.145719	1	13
40	0.742833	0.172638	1	6
60	0.798718	0.18	1	5
80	0.809483	0.189655	1	2
100	0.802589	0.141732	1	4

Table 3. Clustering performances obtained on Reuters-21578 with α=0.7.

Figures 4,5 and 6 show a heuristic model selection strategy adoptable for clustering. Axis x reports the number of cluster while the y axis reports the sum of overall purity and minimum purity. The present strategy selects the model characterized by the maximum of the plotted curves. Such an approach embeds the sum of a global accuracy term (overall purity) and a local term endorsing local accuracy meaning. The best model is the one giving the best balance between local and global accuracy. Note that in this particular case the best model changes accordingly to variations in α. For an infinite number of clusters, this measure will asymptotically converge to 2; however, for a finite number of clusters this measure has local maxima by which a model can be selected. The models selected in this way embed salient local accuracy balanced by a global accuracy metric. In this analysis is implicit that the range of the number of clusters is fixed a priori and is guided, for instance, by the user choice.

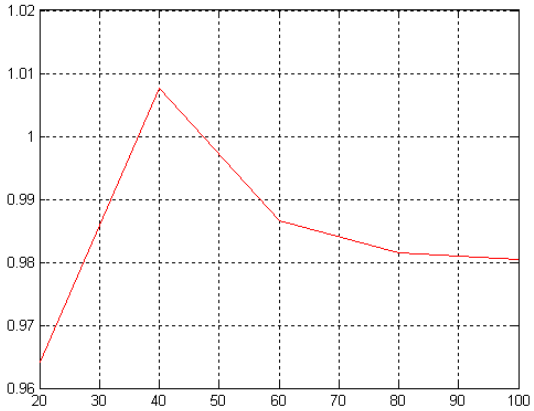

Figure 4. Heuristic Model selection for α=0.3

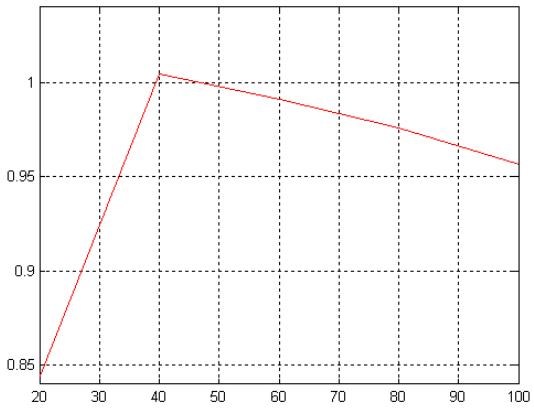

Figure 5. Heuristic Model selection for α=0.5

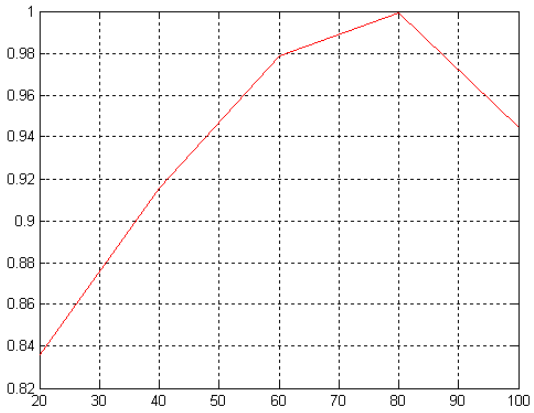

Figure 6. Heuristic Model selection for α=0.7

4.2 Newsgroup 20 dataset

The '20 Newsgroups' corpus includes 20,000 messages collected by using as source 20 different newsgroups. Accordingly, 20 topics have been used to categorize the documents in the corpus. In this case, the documents are almost evenly distributed over the different topics.

The '20 Newsgroups' database provided the second experimental domain for the proposed framework. The experiments involved two different corpora, $\mathcal{D}_{N1}$ and $\mathcal{D}_{N2}$, worked out from such database. Corpus $\mathcal{D}_{N1}$ and corpus $\mathcal{D}_{N2}$ were generated by using the criteria proposed in the work by Jing et al [37]. Thus, $\mathcal{D}_{N1}$ included all the documents (3894 elements) associated to the categories: *comp.graphics*, *rec.sport.baseball*, *sci.space*, and *talk.politics.mideast*; $\mathcal{D}_{N2}$ included all the documents (3929 elements) associated to the categories: *comp.graphics*, *comp.os.ms-windows*, *rec.autos*, and *sci.electronics*.

Table 4 and Table 5 present the results obtained with $\mathcal{D}_{N1}$ and $\mathcal{D}_{N2}$, respectively. In both cases, the setting $\alpha = 0.3$ has been used. Numerical figures show that the system attained with $\mathcal{D}_{N1}$ an overall purity always superior to 0.75; however, clustering performances slightly worsen with corpus $\mathcal{D}_{N2}$. Such a result can be explained by analyzing the characteristics of the two corpora. Corpus $\mathcal{D}_{N1}$ involves categories semantically well separated, while corpus $\mathcal{D}_{N2}$ involves categories that may partially overlap.

Number of clusters	Overall purity	*pur*(*k*) minimum	*pur*(*k*) maximum	Smallest cluster
20	0.755778	0.433409	1	58
40	0.773754	0.314286	1	29
60	0.792244	0.33871	1	9
80	0.819723	0.290323	1	4
100	0.811505	0.311475	1	1

Table 4. Clustering performances obtained on $\mathcal{D}_{N1}$ with α=0.3.

Number of clusters	Overall purity	*pur*(*k*) minimum	*pur*(*k*) maximum	Smallest cluster
20	0.627895	0.303704	1	20
40	0.679817	0.298701	1	14
60	0.691016	0.295775	1	4
80	0.657928	0.265306	1	5
100	0.695597	0.349515	1	5

Table 5. Clustering performances obtained on $\mathcal{D}_{N2}$ with α=0.3.

When comparing these results with those obtained on the same testbed in the work by Jing et al [37] one should take into account the differences in the set up of the two experiments. In that research [37], a modified version of the conventional k-means algorithm, the Entropy Weighting k-Means Algorithm, was used for document clustering. The experiments actually involved a sub-sampled version of the two corpora $\mathcal{D}_{N1}$ and $\mathcal{D}_{N2}$; hence, two corpora including 400 documents each were eventually used to test the proposed clustering scheme, which attained an overall purity larger than 0.88 in both experiments. Figures 7 and 8 show, as per Reuters, the results of the heuristic model selection.

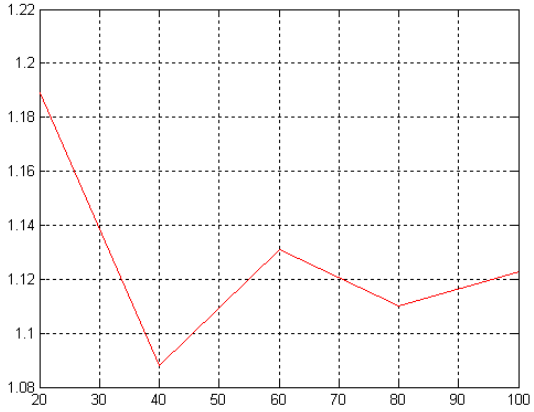

Figure 7. Heuristic Model selection for $\mathcal{D}_{N1}$ and α=0.3

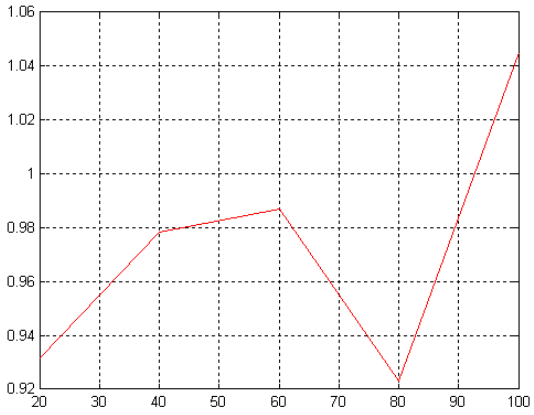

Figure 8. Heuristic Model selection for $\mathcal{D}_{N2}$ and α=0.3

5. Conclusions

Text mining provides a valuable tool to deal with large amounts of unstructured text data. Indeed, in security applications text-mining technologies can help in automating the analysis of existing datasets, with the aim of preventing criminal acts by cataloguing various threads and pieces of information.

Within the text mining environment, document clustering represents one of the most effective techniques to organize documents in an unsupervised manner. A major characteristic of the representation paradigm of text documents is the high dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. Furthermore, the definition of the underlying distance measure between documents is critical for getting meaningful clusters.

The document-clustering approach proposed in this work mostly addresses such issues by combining an effective distance measure and an established pattern-recognition grouping algorithm. Experimental results showed the effectiveness of the proposed approach for two well known texts corpora such as Reuters and Newsgroup 20. Future lines of research may investigate clustering approaches based on semantic information: these, among the prominent, include ontologies and semantic networks.

References

[1] H. Chen, W. Chung, J.J Xu, G. Wang, Y. Qin, M. Chau. "Crime data mining: a general framework and some examples", IEEE Trans. Computer 37, pp.50-56, 2004

[2] J. W. Seifert. "Data Mining and Homeland Security: An Overview". CRS Report RL31798, www.epic.org/privacy/fusion/crs-dataminingrpt.pdf, 2007

[3] J. Mena. *Investigative Data Mining for Security and Criminal Detection*. Butterworth-Heinemann, 2003

[4] D. Sullivan. *Document warehousing and text mining*. John Wiley and Sons, 2001

[5] W. Fan, L. Wallace, S. Rich, Z. Zhang. "Tapping the power of text mining", Comm. of the ACM, 49, pp. 76-82, 2006

[6] R. Popp, T. Armour, T. Senator, K. Numrych. "Countering terrorism through information technology", Comm. of the ACM, 47, pp.36-43, 2004

[7] A. Zanasi (eds.). *Text Mining and its Applications to Intelligence*, CRM and KM. 2nd edition , WIT Press, 2007

[8] Y. Linde, A. Buzo, R.M. Gray. "An algorithm for vector quantizer design", IEEE Trans. Commun. COM-28, pp. 84-95, 1980.

[9] J. Shawe-Taylor, N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

[10] Reuters-21578 Text Categorization Collection. UCI KDD Archive.

[11] The UCI KDD Archive, http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[12] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*, Prentice Hall, 1988

[13] C. D. Manning, P. Raghavan, H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008

[14] S. Ridella, S. Rovetta, and R. Zunino. "Plastic algorithm for adaptive vector quantization", Neural Computing and Applications, 7, pp. 37-51, 1998

[15] V. Vapnik, *The Nature of Statistical Learning Theory,* Springer, 2nd edition, 2000

[16] B. Tang, M. Shepherd, E. Milios, M. Heywood, "Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering". In Proc. of International Workshop on Feature Selection for Data Mining - Interfacing Machine Learning and Statistics, Newport Beach, California, pp. 17-26, 2005

[17] I. S. Dhillon. "Co-clustering documents and words using bipartite spectral graph partitioning", Knowledge Discovery and Data Mining, pp. 269–274, 2001.

[18] D. Hand, H. Mannila, P. Smyth. *Principles of Data Mining*, MIT Press, Cambridge, MA

[19] M.W. Berry, S.T. Dumais, G. W. O'Brien. "Using linear algebra for intelligent information retrieval", SIAM Review, 37, pp. 573–595, 1995

[20] D. Cai, X. He, and J. Han. "Document Clustering Using Locality Preserving Indexing", IEEE Transaction on knowledge and data engineering, 17, pp. 1624-1637, 2005

[21] F. Shahnaz, M. W. Berry , V. Paul Pauca, R. J. Plemmons. "Document clustering using nonnegative matrix factorization", Information Processing and Management, 42, pp. 373–386, 2006

[22] J. Shlens. "A Tutorial on Principal Component Analysis". available at http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf

[23] I. Dhillon, Y. Guan and B. Kulis. "A unified view of Kernel kmeans, Spectral Clustering and Normalized Cuts". UTCS Technical Report #TR-04-25, University

of Texas at Austin, Department of Computer Sciences, Austin, TX 78712, 2005

[24] G. Qian, S. Sural, Y. Gu, S. Pramanik, "Similarity between Euclidean and cosine angle distance for nearest neighbor queries." In Proc. of the 2004 ACM symposium on Applied computing, pp. 1232-1237, 2004

[25] R. Baeza-Yates, B. Ribiero-Neto. *Modern Information Retrieval,* ACM Press, 1999.

[26] L. Jing, M. K. Ng, and J. Zhexue Huang. "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data", IEEE Transactions on knowledge and data engineering, 19, pp. 1026-1041, 2007

[27] M. W. Berry, M. Castellanos. *Survey of Text Mining* II, Springer, 2008.

[28] A. Hotho, A. Nürnberger and G. Paaß. "A brief survey of text mining," LDV Forum - GLDV Journal for Computational Linguistics and Language Technology, 20, pp. 19-62, 2005

[29] Y.-J. Horng, S.-M. Chen, Y.-C. Chang, and C.-H. Lee. "A New Method for Fuzzy Information Retrieval Based on Fuzzy Hierarchical Clustering and Fuzzy Inference Techniques", IEEE Transactions on fuzzy systems, 13, pp. 216-228, 2005

[30] W. –C. Tjhi, L. Chen. "A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data", Fuzzy Sets and Systems, 159, pp. 371–389, 2008

[31] K. M. Hammouda and M. S. Kamel. "Efficient phrase-based document indexing for web document clustering", IEEE Transactions on Knowledge and Data Engineering, 16, pp. 1279–1296, 2004.

[32] H. Chim, X. Deng. "Efficient Phrase-based Document Similarity for Clustering", IEEE transaction on knowledge and data engineering, 20, pp. 1217-1229, 2008.

[33] O. Zamir and O. Etzioni. "Grouper: A Dynamic Clustering Interface to Web Search Results", Computer Networks, 31, pp. 1361-1374, 1999.

[34] G. Salton, A. Wong, L.S. Yang. "A vector space model for information retrieval", Journal Amer. Soc. Inform. Sci., 18, pp. 613-620, 1975

[35] M. Girolami. "Mercer kernel based clustering in feature space", IEEE Trans. Neural Networks, 13, pp. 2780-2784, 2002.

[36] D. Cai, X. He, J. Han. "Document Clustering Using Locality Preserving Indexing", IEEE Transaction on knowledge and data engineering, 17, pp.1624-1637, 2005.

[37] L. Jing, M. K. Ng, and J. Zhexue Huang. "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data", IEEE Transactions on knowledge and data engineering, 19, pp. 1026-1041, 2007

Author Biographies

Sergio Decherchi obtained the "Laurea" degree summa cum laude in Electronic Engineering in 2007 from Genoa University, Italy. Since 2005 he started collaborating with the Department of Biophysical and Electronics Engineering of Genoa University, where he is pursuing a PhD in Electronic Engineering and Computer Science on Machine Learning and Data Mining. His main research areas include: theoretical aspects of Machine Learning, large scale learning algorithms development, semi-supervised learning, dedicated hardware for learning machines and Text Mining.

Paolo Gastaldo obtained the "Laurea" degree in Electronic Engineering and a PhD in Space Sciences and Engineering (2004), both from Genoa University, Italy. Since 2004 he is with the Department of Biophysical and Electronics Engineering of Genoa University, where he is the recipient of a research grant on Intelligent Systems for Visual Quality Estimation sponsored by Philips Research Labs - Eindhoven (NL). His main research area include innovative systems for visual signal understanding, neural network-based methods for nonlinear information processing, and DSP-based architectures for advanced signal interpretation, such as intelligent object tracking for video surveillance and cryptography.

Judith Redi obtained the "Laurea" degree in Electronic Engineering in 2006 from Genoa University, Italy. Her main research areas include the study of cryptographic algorithms and number theory, and the implementation of neural networks models for the objective assessment of visual quality. Since 2007 she joined SEALab as a PhD student in Space Engineering Sciences. She received a grant from Philips Research Labs, Eindhoven, for a five-month visiting period at Philips Labs to enhance research on visual quality assessment, and she's currently collaborating with Philps Research and Delft University of Technology on the use of computational intelligence techniques for the objective assessment of visual quality.

Rodolfo Zunino obtained the "Laurea" degree cum laude in Electronic Engineering from Genoa University in 1985. From 1986 to 1995 he was a research consultant with the Department of Biophysical and Electronic Engineering (DIBE) of Genoa University. He is currently Associate Professor at DIBE, where he teaches Electronics for Embedded Systems and Electronics for Security. His main scientific interests include efficient models for data representation and learning, intelligent electronic systems (VLSI and DSP-based) for neural networks, intelligent systems for security, and advanced methods for multimedia data processing. Rodolfo Zunino coauthored more than 170 refereed papers in International Journals and Conferences. He participated in the Scientific Committees of several International Conferences on neural networks and their applications. He is currently contributing as Associate Editor to the IEEE Transactions on Neural Networks, and is a Senior Member of the IEEE (CIS – Computational Intelligence Society, and Circuits and Systems Society).