A methodology for topographic clustering of structured text documents

Marie-Jeanne Lesot, Delphine Dard, and Florence d'Alché-Buc

Laboratoire d'Informatique de Paris VI,
8, rue du capitaine Scott,
F-75 015 Paris, France
Marie-Jeanne.Lesot@lip6.fr

Abstract. Sets of texts are structured through a more or less refined hierarchy of sections, subsections and paragraphs; this structure contains information that should be exploited to handle these data and in particular, to enrich the comparison of texts, as a complement to the vector description of their contents. We propose a kernel-based methodology that follows this principle for a topographic clustering task and define a *hierarchical kernel* which compares paragraphs using the available hierarchical decomposition and in particular the provided titles.

1   Introduction

Usually, a set of texts is associated with a table of contents which provides titles for the texts and reflects the hierarchical organization proposed by the author; thus it enables the reader to find documents related to his field of interest. It is interesting to complete this table of contents with a transversal representation conveying the similarity between the documents themselves and not the author-defined order: for instance, in the case of texts contained in a history schoolbook, the table of contents mainly reflects the chronological order of the events; an alternative representation may help the pupil to find similar kind of events occurring at different dates.

Organizing texts according to their similarity is a task that has been dealt with by Kohonen *et al.* [6] through Self Organized Maps (SOM) [5] in the case where texts are characterized by the words they contain, in a vectorial description framework. Yet in the case described above, more information is available since the set of texts is structured through a hierarchy of sections, subsections and paragraphs, for which titles are provided; even if the representation only involves the paragraphs themselves, i.e. the deepest level, it appears natural to exploit the whole hierarchy, and in particular the titles: one can consider that the latter rephrase and summarize the context of their subsections at different hierarchical levels, i.e. at different generality or abstration levels; they can contain important key-words that are not repeated in the text itself, and can help characterizing it; thus they can give information about the documents similarity which should complete a classic vectorial description of the texts themselves.

Therefore, the task we consider refers to two problematics, namely topographic clustering and structured data handling. Indeed, the structure information is fundamentally non-vectorial and requires a specific representation. One way to deal with it is to

use the kernel framework [10]: among other, the latter enables to apply classic algorithms defined for vectorial data to other types of data, provided the algorithm can be expressed solely in terms of dot-products. It only requires the user to define a similarity measure which can be expressed as a dot-product in some specific space.

More generally, handling structured data is a currently fast developing research field, which aims at dealing with non-vectorial data such as sequences, trees or graphs, which are often more natural representations for 'real-world' data than vectors. For the text organization task, the structure plays a specific role, which differs from the classic case [3, 2, 8]: it is not associated to each individual datapoint, but it is defined globally; a text is considered as the leaf of a hierarchical tree whose nodes correspond to titles. This implies that one does not compare couples of structured elements, as it is usually done, but couples of vectorial datapoints for which a structural information is available and should be taken into account in the learning procedure.

In this article, we propose a kernel-based methodology to perform topographic clustering on texts for which a hierarchical structure is known: it is based on the definition of an appropriate kernel, called the *hierarchical kernel*, which enables to measure a relevant similarity between such texts. Considering a tree representation of the text set, it defines the similarity between two leaves exploiting the whole hierarchical structure which provides a complementary characterization as compared to the text content itself.

The article is organized as follows: section 2 precises the application setting and the general methodology, section 3 underlines the specificity of the considered structured data as compared to classic structured data handling and presents the kernel proposed to measure the similarity between texts; lastly section 4 contains some preliminary results which justify the methodology and illustrate its relevance.

2   Proposed methodology

In this section, we describe the methodology proposed to provide an organized representation of texts for which a hierarchical decomposition and the associated titles is available; in what follows, we will call *paragraph* the texts to handle, i.e. the texts which must be organized. We successively describe the data encoding, the applied learning algorithm and the evaluation scheme for this unsupervised learning task.

Data encoding The considered dataset consists in a text set for which a hierarchical structure is available, i.e it is a set of paragraphs for which titles of different hierarchical levels are defined, as can be the case for XML texts for instance. It can be represented as a tree whose leaves correspond to paragraphs and whose internal nodes correspond to titles (cf fig. 1): the whole dataset is made of a single tree.

Each part of text, be it a paragraph or a title, is associated with a vectorial description; several encodings can be applied, such as the classic *tfidf* encoding [9] or the *mp-pca* encoding [12], which extracts and represents semantic information. As a result, one gets a tree whose nodes are associated with vectors describing the texts they represent.

The objective of the considered task is then to provide an organized representation of the leaves of this tree using the information contained in the internal nodes.
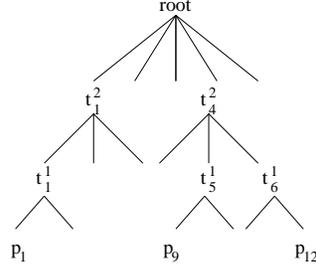
Fig. 1. Tree representation of the whole text set. The leaves, denoted p_i, where i is an identifier, correspond to the paragraphs that are to be represented, the internal nodes, denoted t_k^j where j is associated to a hierarchical level and k is an identifier, correspond to the titles. Each node contains a vectorial description of the associated part of text.

Learning algorithm: kernel-based topographic clustering One method to obtain an organized representation of data is to use the Self Organizing Map (SOM) proposed by Kohonen [5] with a two dimensional output space. Such a topographic clustering algorithm simultaneously identifies subgroups of similar datapoints and preserves information about the relationships between these subgroups: they are associated with positions in the output space such that their relative closeness reflects the similarity of the data they contain.

There exists many variants of SOM (see [7] e.g.), among which a kernel-based algorithm, called Soft Topographic Mapping with Kernel (STMK) proposed by Graepel and Obermayer [4]: it enables to apply the kernel trick [10] to SOM. STMK consists in optimizing the following cost function, which reflects the SOM objective

$$E = \frac{1}{N} \sum_{i=1}^{N} \sum_{r=1}^{K} h_{rg(x_i)} \|\phi(x_i) - w_r\|^2 \text{ with } g(x_i) = \arg\min_s \sum_{t=1}^{K} h_{st} \|\phi(x_i) - w_t\|^2$$

(1)

where x_i, $i = 1..N$ denote the datapoints which belong to $\mathcal{R}^d$, $\phi : \mathcal{R}^d \rightarrow \mathcal{F}$ is a non linear transformation to a high or infinite space, called the *feature space* $\mathcal{F}$; w_r, $r = 1..K$, are the cluster centers, which belong to the feature space and h_{rs} is a $K \times K$ neighborhood matrix which expresses the organization aim by imposing a similarity constraint on neighbor clusters (see [4] for more details), $g(x_i)$ denotes the index of the cluster x_i is assigned to. Optimization is performed through deterministic annealing.

Provided w_r is searched as a linear combination of $\phi(x_i)$ the computations are expressed solely in terms of dot products $< \phi(x_i), \phi(x_j) >$ [4]. Thus, defining a kernel function k such that $k(x_i, x_j) =< \phi(x_i), \phi(x_j) >$, it is possible to optimize E without doing costly calculations in the high dimensional space $\mathcal{F}$ and without expliciting the ϕ function.

Result evaluation As any unsupervised learning task, topographic clustering raises the question of result evaluation, as there is no expected result to compare to. We apply a

quality measure, defined in [7], which takes into account both objectives of topographic clustering, namely the clustering aim and the organization constraint: it combines a compactness criterion with an organization evaluation through an F-measure.

The clustering quality measure , $\tilde{q}$, evaluates the gain in compactness, by comparing the average variance of the identified subgroups to the initial variance of the whole dataset. The organization measure evaluates the correlation between the inter-cluster distances in terms of their position in the two dimensional space (denoted dG) on the one hand and in terms of their centers (denoted dW) on the other hand:

$$\tilde{q} = \frac{\dfrac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{K} p(C_r/x_i)\|\phi(x_i) - \bar{x}_r^{\phi}\|^2}{\dfrac{1}{N}\sum_{i=1}^{N}\|\phi(x_i) - \bar{x}^{\phi}\|^2} \in [0,1]$$

$$\rho = \frac{\sum dGdW - \dfrac{\sum dG \sum dW}{K^2}}{\sqrt{N_G N_W}} \quad \text{and} \quad c = \frac{1+\rho}{2} \in [0,1]$$

$$\mathcal{Q}_b = \frac{(1+b^2)(1-\tilde{q})c}{b^2(1-\tilde{q}) + c} \quad ,$$

where $\bar{x}_r^{\phi}$ is the mean of cluster C_r and $\bar{x}^{\phi}$ that of the whole dataset, computed in the feature space $\mathcal{F}$, and $p(C_r/x_i)$ denotes the probability that the datapoint x_i is assigned to cluster r (it is provided by the deterministic annealing procedure). For any KxK matrix A, $\Sigma A = \sum_{i,j} A_{ij}$ and $N_A = \left(\Sigma A^2 - (\Sigma A/K)^2\right)$.

This measure combines two normalized independant quantities which respectively depend only on the data and the cluster centers. As a consequence of the chosen normalizing factors, these quantities do not depend on the algorithm hyperparameters (size of the map K, neighborhood parameter involved in function h (see eq. (1)) and possibly kernel hyperparameter). The F-measure provides a flexible aggregation mean to compare maps obtained on different datasets for different hyperparameter combinations.

3   Kernel for structured text documents

3.1   Existing approaches

Example of structured data, which correspond to non-vectorial data, such as sequences, trees or graphs, can be drawn from various fields including bioinformatics, which provide sequences (gene expression kinematics for instance) or graphs (chemical molecules), image mining, where images are represented as trees [8, 1] or text mining based on structured formats such as XML or HTML, which also provide tree data [2]. In such cases, each datapoint constitutes a structured element, and the structure is viewed as a discriminative feature which enables to distinguish between underlying classes: for instance Diligenti *et al.* [2] consider a classification task on webpages from a university website and exploit the fact that a page from the *admin* section has a structure which

differs from a *people* section to classify it; likewise, Peura [8] defines a tree matching technique to compute similarities between couples of trees; similarly, many kernels have been defined to compare couples of structured datapoints (see [3] for a survey).

In the case of texts for which a hierarchical structure is known, the structure is to be considered differently: it cannot be seen as an individual feature as it is not associated to each datapoint, but it is defined as an overall structure. Each datapoint corresponds to a leaf of a hierarchical tree whose nodes represent the different titles. Thus there is a single tree for the whole dataset, and the objective is to compare leaves exploiting the hierarchical structure provided.

This can be compared to the point of view considered by Vert [13] to analyse phylogenetic profiles: the handled elements are not structured themselves, they correspond to vectors associated to the set of all leaves of the phylogenetic tree; the kernel-based defined similarity measure exploits the structure information contained in this tree to compare the vectorial datapoints.

3.2   Definition of the hierarchical kernel

Desired semantic Comparing texts exploiting the information about the global hierarchy they belong to is based on the idea that titles can help to compare the texts they refer to: for instance, if two paragraphs handle the same topic with different insights, they may use different words but have similar titles; likewise, if two texts address the same topic, the first one being more general than the second one, it may be useful to compare the first text with the title of the second, to get a same generality level.

Therefore, we define a kernel which computes a weighted average of the similarity between the two texts and their respective titles; the weights indicate that a similarity which only holds at a high generality level is less significant than a low level one.

Formalization To formalize the previous definition, let's denote $\mathcal{S}$ the set of all possible hierarchical levels, encoded as integers increasing with the generality level (1 for the text itself, 2 for its first title, ...); T is the set of paragraphs that are to be compared and $\mathcal{T}$ is the set of all pieces of texts, including the titles; each text is described as a vector, as defined in previous section. Lastly, let ψ_s, for $s \in \mathcal{S}$, be the mapping which associates to a paragraph t the text of its title of level s:

$$\psi_s : T \longrightarrow \mathcal{T}$$
$$t \longmapsto \psi_s(t)$$

Then, for any $(t_1, t_2) \in T^2$ paragraphs couple, the hierarchical kernel is defined as

$$k_\Lambda(t_1, t_2) = \frac{1}{\lambda_{S(t_1, t_2)}} \sum_{s_1, s_2 < S(t_1, t_2)} \lambda_{s_1 s_2} k_e(\psi_{s_1}(t_1), \psi_{s_2}(t_2)) \qquad (2)$$

It is a linear combination of the similarity between couples of hierarchical levels of each text. More precisely,

- k_e is an "elementary" kernel which applies to the vectorial representation of any texts couple. It should be a normalized kernel, i.e. take its values in the interval $[0, 1]$ so that the weighted sum makes sense.

- $\Lambda = (\lambda_{s_1 s_2})_{(s_1,s_2) \in \mathcal{S}^2}$, is a set of weighting factors which express the decrease of importance assigned to a high level similarity: a similarity which only applies to the most general hierarchical levels is less significant than a resemblance at a low level. Therefore $\lambda_{s_1 s_2}$ must be a decreasing function of s_1 and s_2.
- $S(t_1, t_2)$ is defined as the hierarchical level of the least general common title for t_1 and t_2. For instance, it equals 2 for the couple (p_9, p_{12}) of figure 1. It enables to restrict the comparisons between t_1 and t_2 to the titles where they differ. If one takes into account all titles, the defined similarity only reflects the available hierarchy and thus leads to a representation which repeat the table of contents: similar texts are mostly those which share the maximal number of titles, i.e. texts in the same branch of the tree. Now the hierarchy must provide complementary information to the texts similarity and not determine it entirely; otherwise, the representation would be similar to the table of contents, on a two-dimensional space instead of a one-dimensional one.
- $\lambda_{S(t_1,t_2)}$ is a normalizing factor defined as $\lambda_{S(t_1,t_2)} = \displaystyle\sum_{s_1,s_2 < S(t_1,t_2)} \lambda_{s_1 s_2}$.

The computation of the kernel requires to compare all couples of hierarchical levels for both texts, thus its complexity depends on the hierarchical depth of the considered texts. It is to be noted that this parameter is usually quite small and does not imply a high complexity.

In a non structured case, the similarity between two paragraphs t_1 and t_2 is simply computed as $k_e(t_1, t_2)$. The structured case enriches this approach by taking into account crosses comparisons between different hierarchical levels to exploit the information contained in the available titles.

4   Preliminary results on school textbooks

In this section we present the considered application for the proposed methodology and the preliminary tests we carried out to assess the feasability and relevance of the method.

4.1   Characteristics of the datasets

The real application we consider belongs to the framework of a technological transfer RNTL project with the textbook publisher EDITIS: the document sets to handle correspond to school textbooks, their hierarchical structure is the structure of the textbooks. A transversal representation should provide a different insight on the books contents.

We considered two datasets: the first one, $\mathcal{D}_1$, corresponds to the history textbook ; it contains 91 paragraphs, each one having a title, that are organized in 33 subsections and 11 sections. The second one, $\mathcal{D}_2$, contains texts from both history and geography textbooks; it is made of 145 texts, corresponding to 52 subsections and 20 sections. When possible, we enrich titles with introductory paragraphs which also hold for whole subsections. We used a *tfidf* encoding, with a vocabulary of 247 and 400 words for $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively (chosen according to an entropy criterion).

It is to be noted that these real datasets only correspond to small values both for the number of datapoints and the vocabulary size, which makes the learning task difficult.

4.2   Hyperparameter choices

We applied the previously presented methodology to the two datasets, using as elementary kernel a gaussian kernel :

$$k_e(t_1, t_2) = \exp\left(-\frac{\|t_1 - t_2\|^2}{\sigma_k}\right) \quad \text{and} \quad \lambda_{s_1 s_2} = \lambda^{s_1 + s_2}$$

where σ_k and λ are the kernel parameters; in the following, we will denote $k_\Lambda = k_\lambda$.

To implement the decrease of influence associated to high level titles, λ must be taken in the interval $[0, 1]$ so that $\lambda^{s_1 + s_2}$ is a decreasing function of s_1 and s_2. A high value attaches much importance to the structure as it rewards the similarity between high level titles more than smaller λ values.

The STMK algorithm requires to set two hyperparameters, namely the size of the map, K, which corresponds to the maximal number of clusters, and the neighborhood parameter, σ_h, which determines the neighborhood matrix (cf. eq. (1) and [4]). For each dataset, and each (λ, σ_k) value, we select K and σ_h so as to maximize the average global quality $\mathcal{Q}_b$ computed on 20 bootstrap samples.

Lastly, for the evaluation, we make the classic choice $b = 2$ for the F-measure used in the quality measure $\mathcal{Q}_b$, it implies the evaluation rewards more organization than clustering quality, which is relevant for a topographic clustering task (as opposed to a simple clustering task).

4.3   Obtained results

Table 1 shows the results obtained applying the previous choices for various (λ, σ_k); for each σ_k value, the row with no λ value corresponds to the non-structured case. The four first columns present the hyperparameter setting, and the last three the criteria values, respectively for the compactness $\tilde{q}$, which must be minimized, the organization ρ, which must be maximized and the global quality measure $\mathcal{Q}_2$ which must be maximized. The bold figure indicates the best $\mathcal{Q}_2$ value in each case.

It can first be seen that the optimal number of clusters K^* and organization constraint σ_h^* are very stable and do not depend on the kernel parameters σ_k and λ. This indicates that they intrinsically characterize the dataset and give information concerning its compactness and the relation between its subgroups.

One must also notice that the standart deviations computed for the various quantities are quite high; this may be due to the fact that the considered datasets contain few data, therefore the bootstrap process leads to quite different datasets and thus to varying result quality.

For both datasets, it appears that taking into account the structure enables to slightly increase the global quality, and that it mainly influences the organization measure (except for $\lambda = 0.7$ where the quality increase is mainly due to a compactness gain). This effect can also be seen on fig. 2 which gives more details for dataset $\mathcal{D}_1$ and $\sigma_k = 0.5$: it shows the evolution of $\mathcal{Q}_2$ as a function of the neighborhood parameter σ_h, for the non-structured kernel k_e and the hierarchical kernel $k_{0.9}$. It can be seen that for all (K, σ_h) values, $k_{0.9}$ gives better results; the gain appears as significant for large maps, $K = 7$ or 8: this is due to the fact that structure improves the organization quality, which is more sensitive for large maps than for small ones.

Dataset	σ_k	λ	K^*	σ_h^*	$\tilde{q}$	ρ	$\mathcal{Q}_2$
$\mathcal{D}_1$	0.7	-	7	0.16	0.097 ± 0.036	0.393 ± 0.071	0.730 ± 0.035
		0.1	7	0.16	0.091 ± 0.032	0.417 ± 0.066	$\mathbf{0.741} \pm 0.031$
		0.5	7	0.16	0.097 ± 0.041	0.408 ± 0.078	0.736 ± 0.038
		0.9	7	0.16	0.092 ± 0.035	0.400 ± 0.090	0.733 ± 0.043
	0.5	-	8	0.14	0.010 ± 0.013	0.293 ± 0.025	0.695 ± 0.012
		0.1	8	0.14	0.009 ± 0.010	0.313 ± 0.022	0.704 ± 0.011
		0.5	7	0.14	0.099 ± 0.031	0.341 ± 0.080	0.706 ± 0.037
		0.9	7	0.16	0.099 ± 0.037	0.388 ± 0.059	$\mathbf{0.727} \pm 0.029$
	0.1	-	8	0.10	0.049 ± 0.094	0.249 ± 0.050	0.670 ± 0.031
		0.1	8	0.14	0.052 ± 0.046	0.274 ± 0.040	0.682 ± 0.021
		0.5	8	0.14	0.027 ± 0.023	0.289 ± 0.027	0.691 ± 0.013
		0.9	8	0.14	0.020 ± 0.012	0.298 ± 0.030	$\mathbf{0.696} \pm 0.013$
$\mathcal{D}_2$	0.7	-	8	0.16	0.227 ± 0.044	0.277 ± 0.065	0.661 ± 0.031
		0.1	8	0.16	0.224 ± 0.038	0.277 ± 0.083	$\mathbf{0.662} \pm 0.039$
		0.5	8	0.16	0.226 ± 0.048	0.238 ± 0.104	0.644 ± 0.049
		0.9	8	0.16	0.223 ± 0.052	0.256 ± 0.087	0.653 ± 0.041
	0.5	-	8	0.16	0.259 ± 0.048	0.214 ± 0.040	0.629 ± 0.019
		0.1	8	0.16	0.247 ± 0.049	0.226 ± 0.046	0.636 ± 0.022
		0.5	8	0.16	0.245 ± 0.045	0.215 ± 0.050	0.631 ± 0.022
		0.9	8	0.16	0.225 ± 0.048	0.239 ± 0.071	$\mathbf{0.645} \pm 0.033$
	0.1	-	8	0.10	0.230 ± 0.045	0.141 ± 0.065	0.601 ± 0.030
		0.1	8	0.12	0.211 ± 0.027	0.164 ± 0.070	0.614 ± 0.033
		0.5	8	0.12	0.216 ± 0.031	0.166 ± 0.055	0.615 ± 0.027
		0.9	8	0.16	0.227 ± 0.044	0.191 ± 0.052	$\mathbf{0.624} \pm 0.025$

Table 1. For two different databases, results obtained with the values of the map size, K, and the neighborhood parameter, σ_h, that maximize an average of $\mathcal{Q}_2$, with 20 bootstrap samples. Bold numbers correspond to the best $\mathcal{Q}_2$ for each dataset and each σ_k value.

4.4   Discussion

The previous experiments show that the proposed methodology based on the exploitation of hierarchical information is feasible and relevant. We intend to improve the results further thanks to complementary developments.

With the considered real datasets, the difficulty comes from the low number of datapoints which implies a small size of vocabulary: the computed *tfidf* values may have a low characterization power as they imply that two documents are similar if they contain the same words; this can be limiting when considering small vocabularies. Moreover, *tfidf* is here applied to titles, i.e. short texts which usually only contain few words; they are thus encoded by sparse vectors for which a *tfidf* comparison may be too restrictive. Several approaches can be considered to overcome this difficulty.

First, one can modify the elementary kernel k_e which appears in the hierarchical kernel: it has to compare texts of different sizes, for instance when it handles a title and a paragraph, i.e. a much longer text; it is necessary to study its behavior with respect
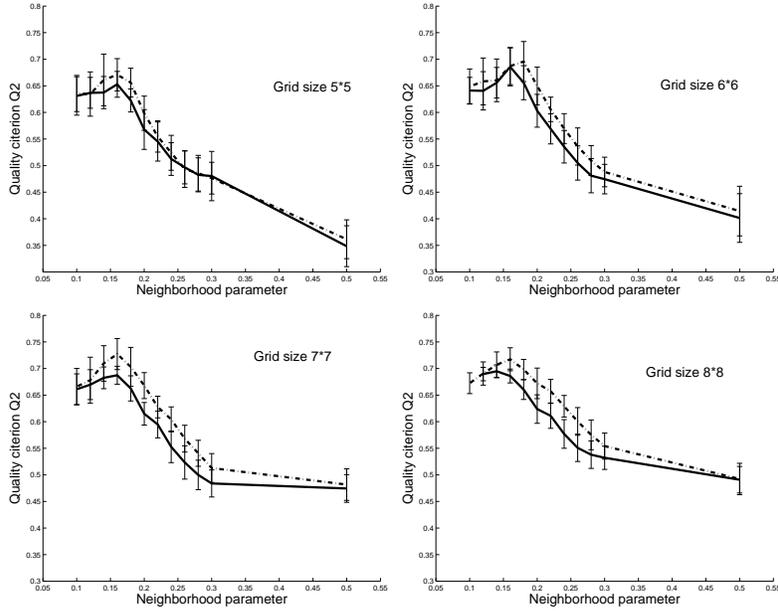
Fig. 2. Evolution of the quality criterion $\mathcal{Q}_2$ for dataset $\mathcal{D}_1$ and $\sigma_k = 0.5$, as a function of the neighborhood parameter σ_h, for the non-structured case (full line) and the structured case (dash-dot line, $\lambda = 0.9$), for different map sizes. The error bars are computed in a bootstrap process with 20 samples.

to these disproportionate data and possibly to modify it so that it takes into account the sparsity of the vectors, or takes a specific form for titles. For instance keeping a Gaussian form for k_e, one could use a width σ_k which depends on the hierarchical level of the compared texts, i.e. depends on the variables s_1 and s_2 using the notations of eq. (2). Another approach is to use as elementary kernel a kernel based on semantic smoothing [11] to incorporate richer information: it exploits a dictionary in order to smooth the text representation so as to take into account prior linguistic knowledge and define semantically close words.

A second method concerns the data representation itself, and aims at using a richer encoding. The idea is to extend the similarity of documents beyond their common words, and for instance consider that two texts are similar if they contain semantically correlated words. This is performed by the *mppca* encoding proposed by Siolas and d'Alché-Buc [12] which defines a generative model of documents based on a generative model of words. It can be interpreted as representing texts with respect to automatically extracted word clusters that correspond to semantic concepts. Like semantic smoothing, this text encoding gives less importance to the occurrences of specific words which can lead to an unstable representation for short texts, and enables to take into account linguistic considerations.

5   Conclusion

We proposed a methodology to handle structured texts documents, where the structure concerns the texts set as a whole and not each datapoint individually, in order to take into account this additional information source. It is exploited through an appropriate kernel which takes into account the titles associated to paragraphs instead of only using *tfidf* representations of texts.

Experiments apply this methodology to real texts data for which the vocabulary is small, as well as the number of documents. The first tests show encouraging results that prove the feasability and relevance of including the information about the hierarchical decomposition of texts sets. Next step consists in improving further this result by enriching the text representation.

Acknowledgements

References

1. M. Diligenti, P. Frasconi, and M. Gori. Image document categorization using hidden tree representations. In *Proc. of the Int. Conf. on Applications of Pattern Recovgnition (ICAPR01)*, pages 147–156. Springer, 2001.
2. M. Diligenti, M. Gori, M. Maggini, and F. Scarselli. Classification of HTML documents by hidden tree-markov models. In *Proc. of the Int Conf. on Document Analysis and Recognition (ICDAR01)*, pages 849–853. IEEE Computer Society, 2001.
3. T. Gärtner. Survey od kernels for structured data. *SIGKDD Explorations*, 5(1):49–59, 2003.
4. T. Graepel and K. Obermayer. Fuzzy topographic kernel clustering. In *Proc. of the 5th GI Workshop Fuzzy Neuro Systems*, pages 90–97. W. Brauer, 1998.
5. T. Kohonen. Analysis of a simple self-organizing process. *Biogical Cybernetics*, 44(2):135–140, 1982.
6. T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3), 2000. Special Issue on Neural Networks for Data Mining and Knowledge Discovery.
7. M.-J. Lesot, F. d'Alché Buc, and G. Siolas. Evaluation of topographic clustering and its kernelization. In *Proc. of the European Conference on Machine Learning*, 2003.
8. M. Peura. The self-organizing map of attribute trees. In *Proceedings of the 9th International Conference on Artifi cial Neural Networks (ICANN'99)*, pages 168–173, 1999.
9. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
10. B. Schölkopf and A. Smola. *Learning with kernels*. MIT Press, 2002.
11. G. Siolas and F. d'Alché-Buc. Support vector machines based on a semantic kernel for text categorization. In *Proc. of ICANN*, 2000.
12. G. Siolas and F. d'Alché-Buc. Mixtures of probabilistic PCAs and Fisher kernels for word and document modeling. In *Proc. of ICANN*. Springer, 2002.
13. J.-P. Vert. A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, 1:1–9, 2002.