International Journal of Pattern Recognition and Artificial Intelligence

Using Sequences of Words for Non-Disjoint Grouping of Documents

Chiheb-Eddine Ben N'Cir
LARODEC, ISG Tunis, University of Tunis, 41 Avenue de la liberté cité Bouchoucha, 2000 le bardo,Tunisie
chiheb.benncir@isg.rnu.tn

Nadia Essoussi
LARODEC, ISG Tunis, University of Tunis, 41 Avenue de la liberté cité Bouchoucha, 2000 le bardo,Tunisie
nadia.essoussi@isg.rnu.tn

Grouping documents based on their textual content is an important application of clustering referred to as text clustering. This paper deals with two issues in text clustering which are the detection of non-disjoint groups and the representation of textual data. In fact, a text document can discuss several topics and then, it must belong to several groups. The learning algorithm must be able to produce non-disjoint clusters and assigns documents to several clusters. Given that text documents are considered as unstructured data, the application of a learning algorithm requires to prepare a set of documents for numerical analysis by using the Vector Space Model (VSM). This representation of text avoids correlation between terms and does not give importance to the order of words in the text. Therefore, we present in this paper an unsupervised learning method, based on the word sequence kernel, where the correlation between adjacent words in text and the possibility of document to belong to more than one cluster are not ignored. In addition, to facilitate the use of this method in text-analytic practice, we present the "DocCO" software which is publicly available. Experiments performed on several text collections show that the proposed method outperforms existing overlapping methods using VSM representation in terms of clustering accuracy.

Keywords: Overlapping Clustering; Document Clustering; Non-disjoint partitioning; Word Sequence Kernel ; Document representation

1. Introduction

Text clustering is a widely used technique in Information Retrieval (IR) to find analogous documents [19], to organize large document collection [11,5,1], to detect duplicate contents and to optimize search engines [16]. This technique aims to group similar documents in the same group or cluster based on their contents, while dissimilar documents must belong to different groups without using any predefined categories. This definition can be a crucial issue in many real life applications of text clustering where a document needs to be assigned to more than one group [18]. This issue arises naturally because a document may discuss several topics and then, must belong to several clusters. For example, a newspaper article concerning the

participation of a soccer player in the release of an action film can be grouped with both of the categories Sports and Movies.

Methods which deal with this challenging issue are referred to as Overlapping Clustering methods [7]. Recent proposed methods use *theoretical* [2,6,10] rather than *heuristic*[13,25] approaches to solve this challenging issue by introducing overlaps in their optimized criteria. These recent methods have been successfully applied in many fields where data require to belong to more than one cluster, but still understudied in Text Clustering. The application of these recent methods to text document needs to prepare the collection of documents for numerical analysis by using Vector Space Model (VSM) representation. This representation of textual documents is based on the assumption that relative position of tokens are irrelevant leading to the loss of correlation with adjacent words and leading to the loss of information regarding word positions. The loss of information and the loss of correlation between adjacent words influence the quality of obtained clusters.

Therefore, we present in this paper a clustering process where the correlation between adjacent words in text and the possibility of document to belong to more than one cluster are not ignored. We propose an overlapping clustering method referred to as KOKM based WSK (Kernel Overlapping k-means based Word Sequence Kernel) which detects relevant and non-disjoint groups in textual data and takes into account information regarding word positions. By introducing overlaps between clusters in the optimized criterion, the proposed method considers textual data as an ordered sequences of words (n-Grams) and uses WSK as similarity measure between documents to avoid high dimensional features of subsequences.

In order to facilitate the use of KOKM based WSK and other overlapping methods in text-analytic practice, we present a Java Graphical User Interface (GUI) called "DocCO" which is publicly available. The main user-related advantages of this GUI are: (a) it can be downloaded freely from the internet, (b) it can easily be used by end-users that are not familiar with software programming, (c) it is easy and flexible in use in that it extends WEKA tool and allows the user to specify different options for the analysis and (d) the results of the clustering can be exported in different formats.

The remainder of this paper is organized as follows: Section 2 presents existing overlapping methods, while Section 3 presents VSM and n-Grams representation of textual documents. The WSK similarity measure between ordered sequences of text is described in Section 4. Then, Section 5 presents the KOKM based WSK method that we propose to detect overlapping clusters from sequences of Text. After that, Section 6 discusses how the "DocCO" software program can be used in practice to perform non-disjoint partitioning of textual documents based on VSM and sequences of words. Experiments on different datasets are described and discussed in Section 7. Finally, Section 8 presents conclusions and future work.

2. Overlapping clustering

The issue of identifying non-disjoint groups has been well studied [7]. First proposed methods are based on *heuristic* approaches which consist either of modifying the clusters resulting from a standard method into overlapping clusters (typically results from k-means or fuzzy-c-means algorithms [13,25]) or in proposing new clustering processes based on intuitive learning such as the CBC (Clustering by Committee) algorithm [17]. These contributions can lead to suitable results in some contexts but they are not predicated on theoretical models and their extension or improvement are limited as a rule [2].

Recent clustering methods propose *theoretical* approaches to solve this challenging issue based on a mathematical formulation of overlaps. These methods are extensions from usual clustering models in which overlaps between clusters are introduced in their optimized criteria. These methods look for optimal overlapping groups and allow observations to belong to more than one cluster. Examples of these methods are MOC (Model based overlapping clustering) [2] and IOMM (Infinite Overlapping Mixture Model) [10] which generalize EM algorithm and OKM (Overlapping-k-means) [6] and KOKMϕ (Kernel Overlapping k-means) [8] which generalize k-means and kernel k-means for overlapping clustering.

These recent methods have been successfully applied in many fields where data require to belong to more than one cluster such as video classification where a video or a film can potentially have multiple genres [20] and emotion detection where a piece of music can evoke several emotions [22,8]. Although the attested performance of these methods to detect relevant overlapping groups, they still understudied in Text Clustering. Therefore, based on OKM and KOKMϕ methods we propose in this paper a clustering method which introduces the possibility of overlaps between clusters in the optimized objective criterion. The proposed method produces hard overlapping partitions while existing techniques are soft (i.e. Fuzzy c-means) requiring a post-processing step to generate overlapping clusters.

3. Preparing documents for numerical analysis : VSM and n-Grams

Textual documents are often considered as unstructured data in which numerical analysis can not be performed. The VSM representation is usually used in Text Clustering to prepare textual documents for a numerical analysis process. In VSM model, each text document is represented by a vector of tokens (words) where the size of vector is determined by the number of different tokens in all documents D. Each documents d_j will be transformed into a vector : $d_j = (w_{1j}, w_{2j}, ..., w_{|T|_j})$, where T is the whole set of terms $T = (t_1, ..., t_{|T|})$ (or tokens) which appears at least once in the corpus ($|T|$ is the size of the vocabulary), and w_{kj} represents the weight (frequency or importance) of the term t_k in the document d_j. Documents whose vectors are close to each others based on tokens frequencies are considered to be similar in content. This representation is based on the assumption that rela-

tive position of tokens has little importance leading to the loss of correlation with adjacent words and leading to the loss of information regarding word positions.

The n-Grams representation of text [24], which is a language independent text representation technique, solves the issue of losing information regarding word positions by considering text document as a sequences of n consecutive characters (syllables or words). The whole set of n-Grams is obtained by the extraction of all possible ordered subsequences of consecutive n characters (syllables or words) along the text. Similarities between documents are measured based on the number of contiguous and non-contiguous subsequences shared between them. This representation leads to high dimensional features of subsequences to represent each text document. This problem is solved in information retrieval tasks by using *Kernel Machines* over sequences of text.

4. Kernel based sequences similarities

Many kernels known as String Kernel have been proposed in literature to solve the problem of high dimensional features in the n-Grams representation of Text by computing only the dot products between the n-Grams of the pair of documents without explicitly computing them for each document. Examples of these kernels are SSK (String Subsequence Kernel) [14] and WSK (Word Sequence Kernel) [3]. SSK measures similarity between two documents based on the number of sequences of characters (implicitly computed) shared between them while WSK measures similarity based on the number of sequences of *words* rather a *characters*. The advantage of using *words* as atomic unit is to keep information regarding word positions in order to maintain linguistic meaning of terms. For example, the term "son-in-law" has a special meaning that can be lost if it is broken. The WSK has also the advantage of reducing the number of features per document because it uses sequences of words rather than sequences of characters. The time complexity of computing WSK kernel between two documents d_1 and d_2 is evaluated to $O(n|d_1||d_2|)$ [3] where n is the length of the used subsequence and $|d_i|$ is the number of words in document d_i.

Let Σ the alphabet which consists in the set of words that exist in all documents, let $S = s_1 s_2 s_3 ... s_{|s|}$ the sequence of words with $|S|$ is the length of S, let $u = s[i]$ a subsequence of S with $s[i] = s_{i_1}..s_{i_j}..s_{i_n}$ where s_{i_1} and s_{i_j} in this subsequence are not necessarily contiguous in S, the feature mapping ϕ for the sentences S in the feature space is given by defining ϕ_u for each $u \in \Sigma^n$ as:

$$\phi_u(S) = \sum_{i:u=s[i]} \lambda^{l(i)}, \tag{1}$$

where $l(i)$ is the length of subsequence $s[i]$ in S with $l(i) = i_n - i_1 + 1$ and λ is the decay factor used to penalize non-contiguous subsequences. These features measure the number of occurrences of subsequence u in the sentences S weighting them according to their lengths. So, given two strings S_1 and S_2, the inner product of the

feature vectors is obtained by computing the sum over all common subsequences:

$$K_n(S_1, S_2) = \sum_{u \in \Sigma^n} \phi_u(s_1)\phi_u(s_2) = \sum_{u \in \Sigma^n} \sum_{i:u=s_1[i]} \sum_{j:u=s_2[j]} \lambda^{l(i)+l(j)}. \quad (2)$$

In order to deal simultaneously with the issue of loosing information regarding word positions and the issue of identifying relevant overlapping clusters, we show in the next section how we introduce WSK as similarity measure to detect overlapping groups from sequential text documents based on the introduction of overlaps in the optimized criterion.

5. Proposed solution : KOKM based WSK

To detect non-disjoint groups from sequential text documents, we propose "KOKM based WSK" using WSK as similarity measure between structured documents. Given a set of N documents $D = \{d_1, d_2, ..., d_N\}$ where each document d_q is defined in the feature space by the sum of u coordinate $\Phi(d_q) = \sum_u \phi_u(d_q)$ which measures the number of occurrences of subsequence u in the document d_q weighted according to its lengths. The aim of the proposed method is to find the optimal assignments matrix $\Pi(N \times C) = \{\pi_1, \pi_2, ..., \pi_C\}$ of documents over C non-disjoints groups. The proposed method consists of the minimization of an objective function defined by the sum of errors E_q local to each document d_q. The sum of local errors over all documents is described by:

$$E_r(\Pi) = \sum_{d_q \in D} E_q = \sum_{d_q \in D} \left\| \Phi(d_q) - \overline{\Phi(d_q)} \right\|^2, \quad (3)$$

where $\overline{\Phi(d_q)}$ is the combination of clusters representatives (typical documents) to which document d_q belongs and is defined by:

$$\overline{\Phi(d_q)} = \frac{\sum_{c=1}^{C} P_{qc} \cdot \Phi(d_{m_c})}{\sum_{c=1}^{C} P_{qc}}, \quad (4)$$

with P_{qc} is a binary variable indicating membership of document d_q in cluster c and d_{m_c} is the typical document of cluster c. Using the Kernel Trick and the Word Sequence Kernel, the objective function is defined as follows:

$$
\begin{aligned}
E_r(\Pi) = & \sum_{d_q \in D} \big[\Phi(d_q)\Phi(d_q) - \frac{2}{L_q}\sum_{c=1}^{C} P_{qc} \cdot \Phi(d_q)\Phi(d_{m_c}) + \\
& (\tfrac{1}{L_q})^2 \sum_{c=1}^{C}\sum_{l=1}^{C} P_{qc}P_{ql} \cdot \Phi(d_{m_c})\Phi(d_{m_l})\big] \\
= & \sum_{d_q \in D} \big[\sum_{u \in \Sigma^n} \phi_u(d_q)\phi_u(d_q) - \frac{2}{L_q}\sum_{c=1}^{C} P_{qc} \cdot \sum_{u \in \Sigma^n} \phi_u(d_q)\phi_u(d_{m_c}) + \\
& (\tfrac{1}{L_q})^2 \sum_{c=1}^{C}\sum_{l=1}^{C} P_{qc}P_{ql} \cdot \sum_{u \in \Sigma^n} \phi_u(d_{m_c})\phi_u(d_{m_l})\big] \\
= & \sum_{d_q \in D} \big[K_n(d_q, d_q) - \frac{2}{L_q}\sum_{c=1}^{C} P_{qc} \cdot K_n(d_q, d_{m_c}) + \\
& (\tfrac{1}{L_q})^2 \sum_{c=1}^{C}\sum_{l=1}^{C} P_{qc}P_{ql} \cdot K_n(d_{m_c}, d_{m_l})\big],
\end{aligned}
\tag{5}
$$

where $L_q = \sum_{c=1}^{C} P_{qc}$ and $K_n(d_i, d_j)$ is the Word Sequence Kernel between document d_i and document d_j as described in Eq. (2).

5.1. *Clustering algorithm of KOKM based WSK*

The minimization of the objective function is performed by iterating two independent steps:

(1) Update of cluster representatives (d_{m_c}).
(2) Multi-assignments of documents to one or several clusters (Π).

The stopping rule of KOKM based WSK algorithm is characterized by two criteria: the maximum number of iterations or the minimum improvement of the objective function between two iterations. The main algorithm of KOKM based WSK is described by Algorithm 1.

Algorithm 1 KOKM based WSK $(D, C, t_{max}, \varepsilon, n, \lambda) \rightarrow \Pi$

Require: D : set of Documents,
t_{max} : maximum number of iterations,
ε: minimal improvement in E_r between two iteration,
C : number of clusters,
n : length of subsequences
λ : decay factor used in WSK.

Ensure:

1: Initialize representatives of clusters over D, Assign documents using "MULTIASSIGN-DOC" and derive value of $E_r(\Pi_0)$ in iteration 0 using Eq. (5).
2: $t = t + 1$
3: Update representatives using Eq. (6).
4: Assign documents to one or several clusters using "MULTIASSIGN-DOC" and derive Π_t.
5: Compute $E_r(\Pi_t)$ using Eq. (5).
6: **if** ($t < t_{max}$ and $E_r(\Pi_{t-1})$ - $E_r(\Pi_t) > \varepsilon$) **then**
7: go to step 2.
8: **else**
9: Return the assignment matrix Π_t.
10: **end if**

Considering the assignments (Π) as fixed, the update of representatives is performed locally for each cluster. Each representative d_{m_c} is defined by the typical document (medoid) in cluster c which minimizes the sum of distances from all documents belonging to the following cluster weighted according to document memberships as described in Eq. (6).

$$\begin{aligned} d_{m_c} &= \min_{q \in \pi_c} \frac{\sum_{j \in \pi_c, j \neq q} w_j . \left\| \Phi(d_q) - \Phi(d_j) \right\|^2}{\sum_{j \in \pi_c, j \neq q} w_j} \\ &= \min_{q \in \pi_c} \frac{\sum_{j \in \pi_c, j \neq q} w_j [K_n(d_q, d_q) - 2.K_n(d_q, d_j) + K_n(d_j, d_j)]}{\sum_{j \in \pi_c, j \neq q} w_j}, \end{aligned} \tag{6}$$

where w_j is a weight assigned to the distance between d_q and d_j depending on the number of clusters to which document d_j belongs to. This weight is more important when assignments of d_j increase in order to reduce its influence in determining the

typical document (document which discuss more than one topic has small probability of being and determining the typical document).

The second step concerns the multi-assignments of documents to one or several clusters. By considering representatives as fixed, we present in the following a generic heuristic "MULTIASSIGN-DOC" which makes the objective function to be minimized and explores the combinatorial sets of possible assignments. The heuristic consists, for each document d_q, in sorting representatives of clusters from closest to farthest, then assigning the document in the order defined while assignment minimizes the local error E_q. The heuristic "MULTIASSIGN-DOC" is described by Algorithm 2.

Algorithm 2 MULTIASSIGN-DOC$(d_q, \{d_{m_1}, ..., d_{m_C}\}, \Pi_q^{old}) \rightarrow \Pi_q$

Require: d_q: Document considered as sequences of words,
$\{d_{m_1}, ..., d_{m_C}\}$: C cluster representatives ,
Π_q^{old}: Old assignments of document d_q.

Ensure:

1: Initialize $\Pi_q = \{d^{\star}_{m_c}\}$ the nearest representative where $d^{\star}_{m_c} = \min_{d_{m_c}} \|\Phi(d_q) - \Phi(d_{m_c})\|^2$ and compute E_q with assignment Π_q.
2: Find the next nearest representative $d^{\star}_{m_c}$ which is not included in Π_q and derive $\Pi_q^{'} = \Pi_q \cup d^{\star}_{m_c}$
3: Compute $E_q^{'}$ with assignment $\Pi_q^{'}$
4: **if** $E_q^{'} < E_q$ **then**
5: $\Pi_q = \Pi_q^{'}$ and return to step 2.
6: **else**
7: compute E_q^{old} with assignment Π_q^{old}.
8: **if** $E_q < E_q^{old}$ **then**
9: return Π_q
10: **else**
11: return Π_q^{old}
12: **end if**
13: **end if**

5.2. *Diagonal dominance problem : Sub Polynomial Kernel as a solution*

The proposed method KOKM based WSK uses the Kernel $K_n(d_i, d_j)$ as similarity measure between documents d_i and d_j. $K_n(d_i, d_j)$ represents the inner product between features of documents d_i and d_j and is evaluated by the sum over all common subsequences weighted according to their frequency of occurrence, lengths and contiguities.

Nevertheless, this definition makes documents having extremely sparse repre-

sentation in the feature space and leads to the overfitting situation when performing the clustering algorithm. The Overfitting situation is reached when all documents are mapped to orthogonal points in the feature space. For overlapping methods, the overfitting situation consists of assigning each document to all clusters. The kernel-based similarity $K_n(d_i, d_j)$ between any pair of distinct documents ($d_i \neq d_j$) will tend to be very small ($K_n(d_i, d_j) \cong 0$) with respect to the self-similarity of documents ($K_n(d_i, d_i) = 1$), especially for larger values of n. The Gram matrix tends to be nearly diagonal because off-diagonal entries are very small and diagonal entries are equal to 1 meaning that all documents are mapped to nearly orthogonal points.

In order to overcome this problem, we propose to integrate Sub-Polynomial Kernels [12] within KOKM based WSK. Sub-Polynomial Kernels are used in Kernel machines to avoid the diagonal dominance in the Gram matrix. Given a positive kernel $K(x_i, x_j)$, the Sub-Polynomial Kernel is defined as :

$$K^{sp}(x_i, x_j) = (K(x_i, x_j))^p = \langle \phi(x_i), \phi(x_j) \rangle^p, \tag{7}$$

where $p \in [0, 1]$ is the degree of the Sub-Polynomial Kernel. As the value of p decreases ($p \longmapsto 0$), the ratio of diagonal entries to off-diagonal entries in the Gram matrix decreases ($ratio \longmapsto 1$). Therefore, to deal with the diagonal dominance problem, we define a new Kernel based similarity $K_n^{sp}(d_i, d_j)$ between any pair of documents d_i and d_j which is described by :

$$K_n^{sp}(d_i, d_j) = (K'_n(d_i, d_j))^p, \tag{8}$$

where $K'_n(d_i, d_j)$ is the normalization of $K_n(d_i, d_j)$ to prevent influence of weighting according to the length of subsequences and is described by:

$$K'_n(d_i, d_j) = \frac{K_n(d_i, d_j)}{\sqrt{K_n(d_i, d_i) K_n(d_j, d_j)}}. \tag{9}$$

6. DocCO: Java GUI for non-disjoint partitioning of documents

In order to facilitate the task of non-disjoint grouping of documents for information retrieval practitioner, we present a java Graphical User Interface (GUI) for learning from vectorial and sequential textual documents which can be downloaded at "https://sourceforge.net/projects/documentclustering/". The application is available under the GNU GPL licence. It requires Java version 1.6. This application was developed within WEKA machine learning toolkit. This choice was made in order to take the advantages of the vast resources of WEKA on pre-processing textual data and the familiarity of many machine learning researchers and practitioners with this tool. The proposed GUI offers the possibility to make non-disjoint clustering of documents using both vectorial and sequential representation. The GUI can easily be run by downloading the folder "DocCO.rar" and by double clicking on the file "DocCO.jar". The GUI can also be run from the command line by typing: `java -jar ''DocCO.jar''`.

The main interface of the application is reported in Figure 1. We present below the different functionalities offered by the GUI.

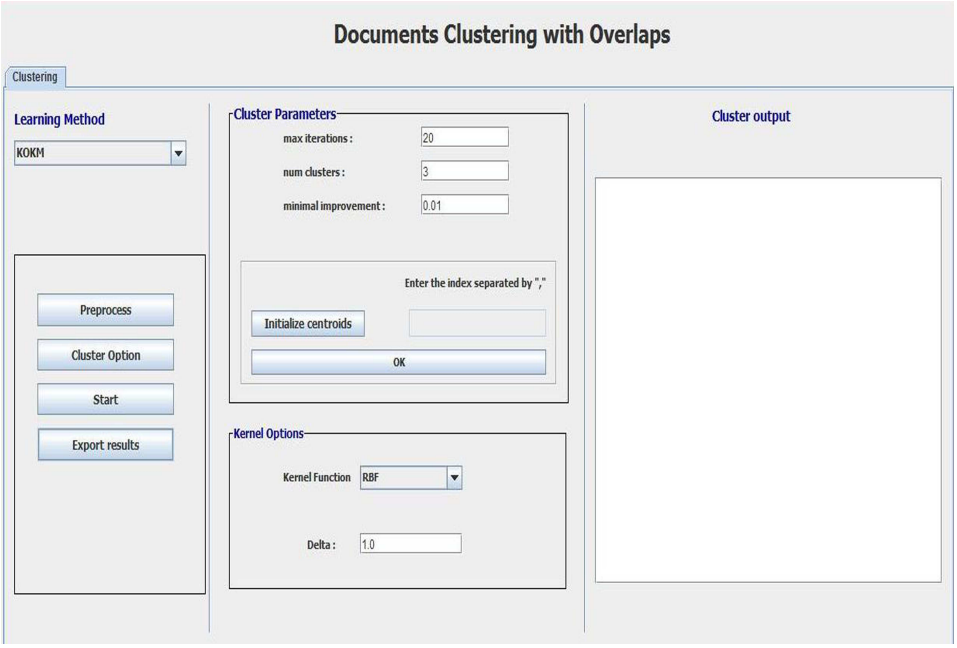


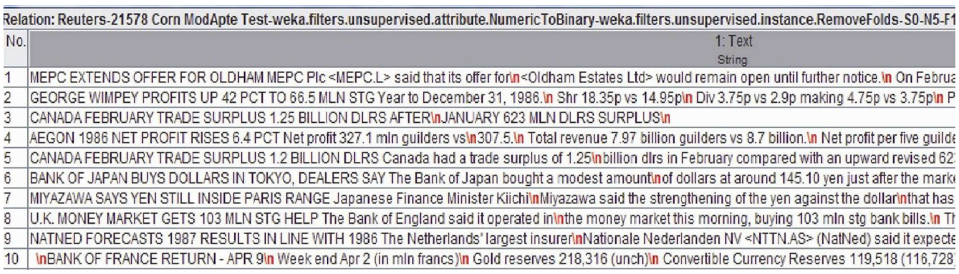
Fig. 1. Main interface of DocCO

6.1. *Loading and pre-processing documents*

The loading functionality, which can be activated using the button "Preprocess", offers many standard loader defined in WEKA such as ARFF loader, CSV loader and the text directory loader. All data format supported by WEKA can be used in DocCO. Data can be loaded from files, from databases or from specified URL. All the pre-processing techniques implemented in WEKA (stemming, stop word removal, tokenizer, TF-IDF transformation, etc.) can be used before performing the learning. For using the word sequence approach, data should be loaded as a string attribute containing all the textual description of the document. Figure 2 shows an example of a set of documents from Reuters dataset which can be learned using the word sequence approach.

6.2. *Grouping documents*

Grouping documents can be realized using several learning methods. The listbox "Learning Method" allows to choose the learning algorithm to be performed for grouping documents. Four methods can be chosen: KOKMϕ, OKM, k-means which are based on the VSM representation of text and KOKM based WSK which uses the word sequence representation. We note that all these learning methods lead to non-disjoint partitioning, except k-means which is given as baseline. For the implementation of WSK, we used a recursive definition [14] based on the dynamic



Relation: Reuters-21578 Corn ModApte Test-weka.filters.unsupervised.attribute.NumericToBinary-weka.filters.unsupervised.instance.RemoveFolds-S0-N5-F1

No.	1: Text String
1	MEPC EXTENDS OFFER FOR OLDHAM MEPC Plc <MEPC.L> said that its offer for\n<Oldham Estates Ltd> would remain open until further notice.\n On Februa
2	GEORGE WIMPEY PROFITS UP 42 PCT TO 66.5 MLN STG Year to December 31, 1986.\n Shr 18.35p vs 14.95p\n Div 3.75p vs 2.9p making 4.75p vs 3.75p\n P
3	CANADA FEBRUARY TRADE SURPLUS 1.25 BILLION DLRS AFTER\nJANUARY 623 MLN DLRS SURPLUS\n
4	AEGON 1986 NET PROFIT RISES 6.4 PCT Net profit 327.1 mln guilders vs\n307.5.\n Total revenue 7.97 billion guilders vs 8.7 billion.\n Net profit per five guilde
5	CANADA FEBRUARY TRADE SURPLUS 1.2 BILLION DLRS Canada had a trade surplus of 1.25\nbillion dlrs in February compared with an upward revised 62
6	BANK OF JAPAN BUYS DOLLARS IN TOKYO, DEALERS SAY The Bank of Japan bought a modest amount\nof dollars at around 145.10 yen just after the mark
7	MIYAZAWA SAYS YEN STILL INSIDE PARIS RANGE Japanese Finance Minister Kiichi\nMiyazawa said the strengthening of the yen against the dollar\nthat has
8	U.K. MONEY MARKET GETS 103 MLN STG HELP The Bank of England said it operated in\nthe money market this morning, buying 103 mln stg bank bills.\n Th
9	NATNED FORECASTS 1987 RESULTS IN LINE WITH 1986 The Netherlands' largest insurer\nNationale Nederlanden NV <NTTN.AS> (NatNed) said it expecte
10	\nBANK OF FRANCE RETURN - APR 9\n Week end Apr 2 (in mln francs)\n Gold reserves 218,316 (unch)\n Convertible Currency Reserves 119,518 (116,728

Fig. 2. Loading textual data as a string attribute in DocCO for learning groups using the word sequence approach

programming technique. The advantage of this implementation is to reduce the time complexity and to perform WSK without explicitly extracting word sequences. We used an online implementation of all the kernel functions to avoid the pre-computing of the kernel matrix.

Before starting the learning process, some of cluster parameters and kernel options must be configured. We give in the following a short description of these parameters:

- **Cluster Parameters**
  - Max iterations: the maximal number of iterations of the main algorithm
  - Num cluster: the number of clusters to be considered
  - Minimal improvement: the minimal improvement in the objective function between two iterations which is used for the convergence of the method. If the improvement is less than the specified value in this text field, the used learning method returns clusters obtained at the final iteration. By default, this value is initialized as 0.01
  - Initialize centroids: this functionality can be used to manually configure the set of documents which will be used as initial representative of clusters known as seeds. Indexes of documents to be considered as typical documents should be entered separated by comma and must exactly match with the number of clusters specified in "Num cluster". For example, for initializing representative of 3 clusters using the first three documents in the corpus we use "1,2,3". If this functionality is not activated, typical documents used as initial clusters' representative are chosen randomly [4].
- **Kernel options**
  - Kernel function: this parameter must be configured for methods which incorporate kernels. Many standard Kernel functions can be chosen in the listbox responding to the data representation's model. "Linear", "Polynomial" and "RBF" kernels can be configured within KOKMϕ if VSM representation of text is used. However, if textual data are loaded as a string attribute, WSK kernel must be configured within KOKM based

WSK
- Value of the kernel parameter: this option indicates the value of the kernel function's parameter. For example, the degree "d" for Polynomial kernel, the size of frame "σ" for RBF kernel and the length of the sequence of words "n" for WSK kernel.

Once parameters and attributes are configured, the learning process can be performed by activating the functionality "Start". If "Class" attributes are loaded within the collection of data or any other attributes which should not be included in the learning process, the GUI allows to ignore these attributes by using the functionality "Cluster option".

6.3. *Outputs and export of results*

The resulting partitioning is visualized in the tab "Cluster output". This tab firstly reports information regarding data such as the number of textual documents and the number of the considered attributes. Then, it reports information regarding the learning process such as the number of iterations of the main algorithm, the value of the optimized criterion and the returned typical documents. Finally, it reports the final binary assignment matrix where rows represent documents, columns represent clusters and the internal value "1" indicates that document i belongs to the respective cluster. Figure 3 shows an example of clusters output obtained using KOKM based WSK with 5 clusters in a collection of Reuters containing 50 documents.

In order to enable further processing of the outputs, the final binary matrix can be exported and saved in a separated file using the functionality "Export results".

7. Experiments and Discussions

7.1. *Evaluation methodology*

Clustering validation is known to be a difficult task in pattern recognition, mainly because of the vagueness of the definition of a "good clustering". In addition, most of the validity measures traditionally used for clustering assessment are inappropriate for overlapping clustering. Therefore, to evaluate the quality of obtained groups of documents, we used an extension of external validation measures (Precision, Recall and F-measure) for multi-labeled data based on the Label Based Evaluation methodology as described by Tsoumakas [21]. These validation measures attempt to estimate whether the prediction of categories is correct with respect to the underlying true categories in the data.

Given a set of documents $D = \{d_1, ..., d_N\}$ and two partitions over D to compare, $C = \{c_1, ..., c_k\}$ a partition of D into k classes (true labels), and $R = \{r_1, ..., r_k\}$ a partition of D into k clusters (R is defined by the clustering algorithm), we consider the following :

- True positive TP_i : the number of documents in r_i that exist in c_i

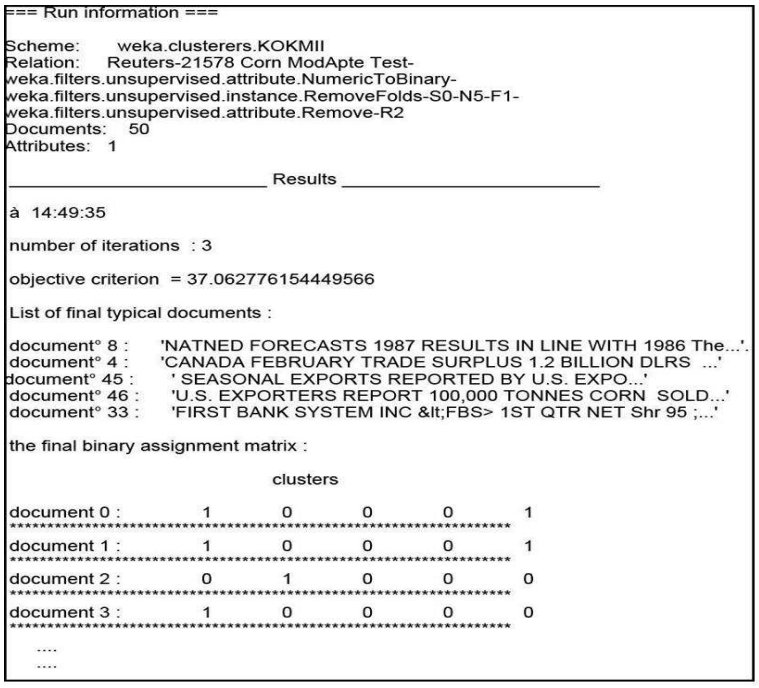


Fig. 3. Example of clusters output obtained using KOKM based WSK with 5 clusters on a collection of Reuters containing 50 documents.

- False negative FN_i : the number of document in c_i that do not exist in r_i
- False positive FP_i : the number of documents in r_i that do not exist in c_i

The used external validation measures are computed for each label i as follows:

$$Precision_i = \frac{TP_i}{TP_i+FP_i}$$
$$Recall_i = \frac{TP_i}{TP_i+FN_i}$$
$$F-measure_i = \frac{(2*Recall_i*Precision_i)}{(Recall_i+Precision_i)}.$$

The computation of external validation measure for all labels is achieved using macro-averaging technique which is usually used in Information Retrieval tasks to evaluate clustering results when the number of classes is not large [23].

Experiments are conducted on two overlapping textual datasets which are respectively Reuters [a] and Ohsumed [b] datasets. The first dataset contains 21578 English newspaper documents. Each document is labeled by one or several labels from a set of 114 categories. We used different collections [c] which are composed of

[a]cf.http://kdd.ics.uci.edu/databases/reuters-transcribed/reuters-transcribed.html
[b]cf.http://disi.unitn.it/moschitti/corpora/ohsumed-first-20000-docs.tar.gz
[c]Collections are manually built over 10 categories where documents which belong to more than one category was kept leading to overlap rate between 1.2 and 1.5 for all collections

100, 500, 800 and 2000 documents. Each document in the used collections belongs to one or several categories from a set of 10 categories. The second dataset is a set of different references from the On-Line Medical Information database (MEDLINE), consisting of titles and abstracts from 270 medical journals over five-year period. We extract a collection of Ohsumed composed of 200 documents distributed over 5 categories.

7.2. *Empirical results*

Experiments are performed on computer with 4 GB RAM and 2.1 GHZ Intel Core 2 duo processor. Data are pre-processed by removing stop words. The VSM representation of each dataset is built using the "DocCO" where occurrences of words are computed using the $TF - IDF$ technique. Table 1 and Table 2 report average scores and standard deviations of Precision, Recall and F-measure on ten runs using k-means, OKM and KOKMϕ methods based on VSM representation compared to the proposed method KOKM based WSK. For KOKMϕ method, we studied its performance using different types of kernels (Linear, Polynomial and RBF) while for KOKM based WSK, we used $n = 2$ and $n = 3$ the length of word sequences, $\lambda = 0.9$ the value of the decay factor and $p = 0.05$ the power of the sub polynomial kernel. For each run, all methods are computed with same initialization of seeds to guarantee that all methods have the same experimental conditions. Values in bold correspond to the best obtained scores.

As reported in Table 1 and Table 2, KOKM based WSK builds in almost collections higher quality clusterings than all the other methods, according to the F-measure. For summarizing the above results, we report the statistical significance matrix for F-measure values obtained by each method as shown in Table 3. In this table, the symbols ">>" ("<<") indicates that F-measure values obtained by the method of the row are significantly better (worse) than the values obtained by the method of the column; the symbol ">" ("<") indicates that the relation is not significant. For testing the statistical significance we used the MannWhitney U-test [15] with a 90% of confidence. This non-parametric test is usually used to test whether one of two random variables is stochastically larger than the other[9].

As it can be seen from Table 3, KOKM based WSK significantly outperforms the other algorithms used in the comparison, in terms of clustering quality. In fact, obtained F-measure with overlapping methods outperforms F-measure obtained with hard k-means. The improvement is induced by a remarkable improvement of Recall. Results obtained with hard k-means are characterized by high value of Precision and low value of Recall. Hard k-means fails to detect groups of document when the dimensionality increases which explains the low value of Recall when the number of documents increases as shown on Reuters-2000 (Recall obtained with k-means is 0.132).

Table 1. Comparison of the performance of KOKM based WSK versus existing VSM based-methods on different collections of Reuters dataset.

Dataset	Methods	Precision	Recall	F-measure
Reuters-100	k-means	**0,570±0,02**	0,262±0,03	0,351±0,02
	Fuzzy-c-means ($\theta = \frac{1}{k}$)	**0,562±0,01**	0,283±0,04	0,359±0,02
	OKM	0,275±0,01	**0,968±0,03**	0,429±0,01
	KOKMϕ (Linear)	0,275±0,01	0,955±0,04	0,427±0,02
	KOKMϕ (Polynomial)	0,275±0,01	0,955±0,04	0,427±0,01
	KOKMϕ (RBF σ=10^{10})	0,275±0,01	0,955±0,05	0,427±0,02
	KOKM based WSK (n=2)	**0,425±0,04**	0,717±0,12	**0,534±0,06**
	KOKM based WSK (n=3)	**0,436±0,06**	0,721±0,13	**0,540±0,09**
Reuters-500	k-means	0,427±0,03	0,132±0,10	0,201±0,07
	Fuzzy-c-means ($\theta = \frac{1}{k}$)	0,432±0,04	0,142±0,10	0,223±0,07
	OKM	0,308±0,01	0,136±0,03	0,188±0,01
	KOKMϕ (Linear)	0,162±0,01	0,314±0,04	0,214±0,02
	KOKMϕ (Polynomial)	0,438±0,01	0,273±0,04	0,336±0,01
	KOKMϕ (RBF σ=10^{10})	0,388±0,01	0,141±0,05	0,207±0,02
	KOKM based WSK (n=2)	0,283±0,04	0,759±0,12	**0,410±0,04**
	KOKM based WSK (n=3)	0,200±0,04	0,466±0,12	0,280±0,04
Reuters-800	k-means	0,470±0,03	0,135±0,36	0,214±0,04
	Fuzzy-c-means ($\theta = \frac{1}{k}$)	0,470±0,03	0,135±0,36	0,214±0,04
	OKM	0,122±0,01	0,583±0,03	0,202±0,01
	KOKMϕ (Linear)	0,170±0,01	**0,613±0,04**	0,267±0,02
	KOKMϕ (Polynomial)	0,122±0,01	0,583±0,04	0,202±0,01
	KOKMϕ (RBF σ=10^{10})	0,457±0,01	0,144±0,05	0,219±0,02
	KOKM based WSK (n=2)	0,334±0,04	**0,620±0,12**	**0,434±0,04**
	KOKM based WSK (n=3)	0,324±0,04	0,530±0,12	**0,402±0,04**
Reuters-2000	k-means	**0,520±0,04**	0,132±0,09	0,216±0,06
	Fuzzy-c-means ($\theta = \frac{1}{k}$)	**0,520±0,04**	0,132±0,09	0,216±0,06
	OKM	0,183±0,01	0,316±0,03	0,232±0,01
	KOKMϕ (Linear)	0,128±0,01	0,391±0,04	0,193±0,02
	KOKMϕ (Polynomial)	0,150±0,01	0,321±0,04	0,205±0,01
	KOKMϕ (RBF σ=10^{10})	0,112±0,01	0,362±0,05	0,171±0,02
	KOKM based WSK (n=2)	0,345±0,04	**0,487±0,12**	**0,404±0,04**
	KOKM based WSK (n=3)	0,257±0,04	**0,568±0,12**	**0,354±0,04**

Table 2. Comparison of the performance of KOKM based WSK versus existing VSM based-methods on a collection of Ohsumed dataset.

Dataset	Methods	Precision	Recall	F-measure
Ohsumed-200	k-means	**0,450±0,03**	0,180±0,36	0,252±0,04
	Fuzzy-c-means ($\theta = \frac{1}{k}$)	**0,455±0,02**	0,188±0,28	0,258±0,13
	OKM	0,274±0,03	0,799±0,36	0,396±0,04
	KOKMϕ (Linear)	0,297±0,10	0,798±0,36	0,417 ±0,11
	KOKMϕ (Polynomial)	0,297±0,10	0,798±0,35	0,417±0,10
	KOKMϕ (RBF σ=10^8)	0,262±0,04	0,835±0,40	0,385±0,03
	KOKM based WSK (n=2)	**0,324±0,03**	0,694±0,08	**0,441±0,02**
	KOKM based WSK (n=3)	**0,319±0,03**	0,709±0,08	**0,440±0,02**

Table 3. Statistical significance matrix for F-measure values.

Method	k-means	Fuzzy-c-means	OKM	KOKMϕ (RBF)	KOKMϕ (Polynomial)	KOKM based WSK
k-means	-	<	<	<<	<	<<
Fuzzy-c-means	>	-	<	<	<<	<<
OKM	<	<	-	<	<	<<
KOKMϕ (RBF)	>	>	>	-	<	<<
KOKMϕ (Polynomial)	>>	>>	>>	>	-	<<
KOKM based WSK	>>	>>	>>	>>	>>	-

The F-measure obtained with KOKM based WSK is characterized by a high value compared to overlapping methods based the VSM representation. The improvement of F-measure is induced by the improvement of Precision. For example, on Reuters-100 dataset, the obtained Precision using KOKM based WSK is 0.425 while using KOKMϕ and OKM methods the max obtained Precision is 0.275. Recalls obtained with KOKMϕ and OKM methods are characterized by high values (Recall obtained with OKM on Reuters-100 dataset is 0.968). These high values of recall is explained by the way that OKM and KOKMϕ assign documents to all clusters because of the high dimensionality of data. For example, on Reuters-100 dataset where the dimensionality of the VSM matrix is very sparse (1482 words) OKM and KOKMϕ assign each document to practically all clusters. This problem is solved when using KOKM based WSK which explains the large improvement in terms of Precision.

Obtained results demonstrate that maintaining order in text improves clustering accuracy compared to VSM representation. The frequent word sequences can provide compact and valuable information about documents structures. However,

we notice that using sequences of words would take more computational time than using VSM representation as described in Table 4 where the runtime required for KOKM based WSK is large compared to the runtime of KOKMϕ.

Table 4. Comparison of the runtime of KOKM based WSK and KOKMϕ on different collections of text.

Dataset	**#Documents**	**#Clusters**	**KOKM based WSK**	**KOKMϕ**
Reuters-100	100	10	190 Seconds	**5** Seconds
Reuters-500	500	10	1205 Seconds	**40** Seconds
Reuters-2000	2000	10	3125 Seconds	**61** Seconds
Ohsumed	200	5	123 Seconds	**12** Seconds

7.3. *Sensitivity of KOKM based WSK to predefined parameters*

We study in this section the sensitivity of KOKM based WSK to parameters λ (value of the decay factor), n (length of the subsequences) and p (power of the sub polynomial kernel). All these parameters should be initialized before performing KOKM based WSK. The first parameter $\lambda \in [0, 1]$ is used to penalize non-contiguous subsequences. When λ is near to 0, the non-contiguous word subsequences are considered *dissimilar* and then, the gap is more penalized. Table 5 reports obtained results using different values of λ used within KOKM based WSK on Reuters and Ohsumed datasets. Obtained F-measures are little sensitive to λ. These results prove that locality does not have an important impact on the performance of the proposed method when applied to text clustering.

To study the sensitivity of KOKM based WSK to both parameters n and p, we fix $\lambda = 0.8$ and we perform experiments with different values of these parameters. Table 6 reports the average of obtained results on Reuters and Ohsumed datasets over ten runs with same initialization of cluster representatives. We notice a high sensitivity of KOKM based WSK to parameters n and p. For example, on Reuters dataset, obtained values of Precision lie between 0.274 and 0.440 and obtained values of Recall lie between 0.681 and 0.968. Concerning the impact of n, obtained F-measures are slightly improved when this parameter increases, until $n = 4$ where F-measures are reduced as shown in Figure 4. These results can be explained by the difficulty to find similar subsequences of words between textual documents since the length of subsequences becomes larger than 3. In this case, documents are considered all different which explains the high value of Recall and the low value of Precision.

Concerning the impact of p, the obtained Precision increases and obtained Recall decreases when n decreases leading to the improvement of F-measure on both

Table 5. The impact of varying the decay factor λ when using KOKM based WSK.

Dataset		Precision	Recall	F-measure
Reuters	λ=0,1	0,458±0,045	0,698±0,023	**0,553±0,037**
	λ=0,3	0,443±0,027	0,664±0,096	0,531±0,050
	λ=0,5	0,434±0,060	0,705±0,093	0,536±0,058
	λ=0,7	0,431±0,042	0,708±0,103	0,535±0,063
	λ=0,9	0,440±0,065	0,705±0,054	0,540±0,031
Ohsumed	λ=0,1	0,320±0,021	0,587±0,070	0,413±0,011
	λ=0,3	0,325±0,022	0,616±0,042	**0,433±0,018**
	λ=0,5	0,323±0,021	0,587±0,035	0,416±0,017
	λ=0,7	0,310±0,014	0,614±0,080	0,411±0,028
	λ=0,9	0,307±0,014	0,695±0,050	**0,426±0,014**

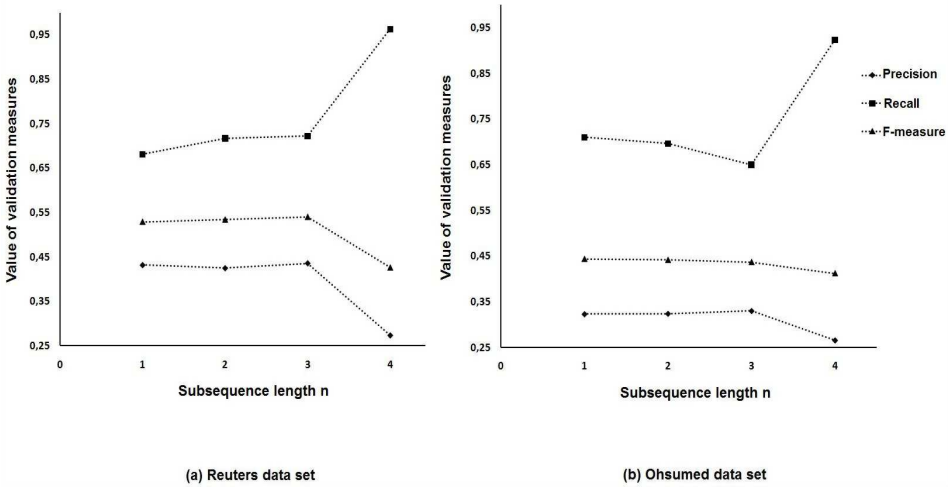


Fig. 4. The impact of varying subsequences length n on the performance of KOKM based WSK

Reuters and Ohsumed datasets as illustrated in Figure 5. The high values of Recall (Recall $\longmapsto$ 1) which coincides with high values of p ($p \longmapsto 1$), are induced by the off-diagonal problem in the Gram Matrix. This problem is solved when using small values of p ($p \longmapsto 0$). For example, Recall is reduced from 0.968 to 0.722 when the value of p varies from 0.6 to 0.05.

8. Conclusion

We deal in this paper with the issue of identifying overlapping groups from sequences of texts. We have proposed the KOKM based WSK method which is able

Table 6. The impact of varying the subsequence length n and the power p on the performance of KOKM-based-WSK.

Dataset	Length	Power	Precision	Recall	F-measure
Reuters dataset	n=1	p=0,05	0,432±0,107	0,681±0,150	0,529±0,126
	n=1	p=0,20	0,401±0,103	0,689±0,129	0,503±0,114
	n=1	p=0,40	0,410±0,028	0,750±0,013	0,530±0,023
	n=1	p=0,60	0,383±0,029	0,769±0,018	0,511±0,027
	n=2	p=0,05	0,425±0,032	0,717±0,045	0,534±0,037
	n=2	p=0,10	**0,440±0,064**	0,705±0,054	**0,540±0,030**
	n=2	p=0,20	0,388±0,042	0,774±0,033	0,516±0,043
	n=2	p=0,40	0,331±0,061	0,884±0,037	0,480±0,066
	n=2	p=0,60	0,277±0,006	0,787±0,509	0,391±0,113
	n=3	p=0,05	0,436±0,086	0,722±0,039	**0,540±0,080**
	n=3	p=0,10	0,426±0,072	0,729±0,058	0,537±0,065
	n=3	p=0,20	0,358±0,014	0,811±0,021	0,497±0,014
	n=3	p=0,40	0,274±0,008	0,963±0,037	0,427±0,012
	n=3	p=0,60	0,274±0,010	**0,968±0,032**	0,427±0,015
Ohsumed dataset	n=1	p=0,05	0,323±0,027	0,710±0,041	**0,444±0,018**
	n=1	p=0,10	0,319±0,027	0,694±0,023	0,437±0,029
	n=1	p=0,20	0,304±0,022	0,725±0,040	0,428±0,014
	n=1	p=0,40	0,291±0,015	0,763±0,020	0,421±0,012
	n=1	p=0,60	0,280±0,001	0,860±0,012	0,423±0,000
	n=2	p=0,05	0,324±0,014	0,696±0,070	0,442±0,018
	n=2	p=0,10	0,308±0,012	0,696±0,038	0,426±0,015
	n=2	p=0,20	0,279±0,012	0,760±0,046	0,408±0,014
	n=2	p=0,40	0,253±0,004	0,978±0,038	0,402±0,007
	n=2	p=0,60	0,249±0,006	**0,985±0,025**	0,397±0,010
	n=3	p=0,05	**0,330±0,017**	0,656±0,036	0,437±0,023
	n=3	p=0,10	0,319±0,014	0,709±0,004	0,440±0,012
	n=3	p=0,20	0,265±0,003	0,923±0,033	0,412±0,000
	n=3	p=0,40	0,249±0,006	**0,985±0,025**	0,397±0,010
	n=3	p=0,60	0,249±0,006	**0,985±0,025**	0,397±0,010

to detect non-disjoint groups from sequential textual documents based on WSK as similarity measure. Empirical results on several textual collections show that detecting overlapping groups by considering text as a sequence of words improves quality of clustering compared to the VSM representation of text.

To easily interpret results after classification we consider a hard assignment $P_{qc} \in \{0, 1\}$ of a document q to cluster c. Each document would be a member or not of one or several groups. However, one could add a probabilistic or fuzzy assignments of the document to model its membership to each cluster.

The proposed method can be applied for many others application domains where

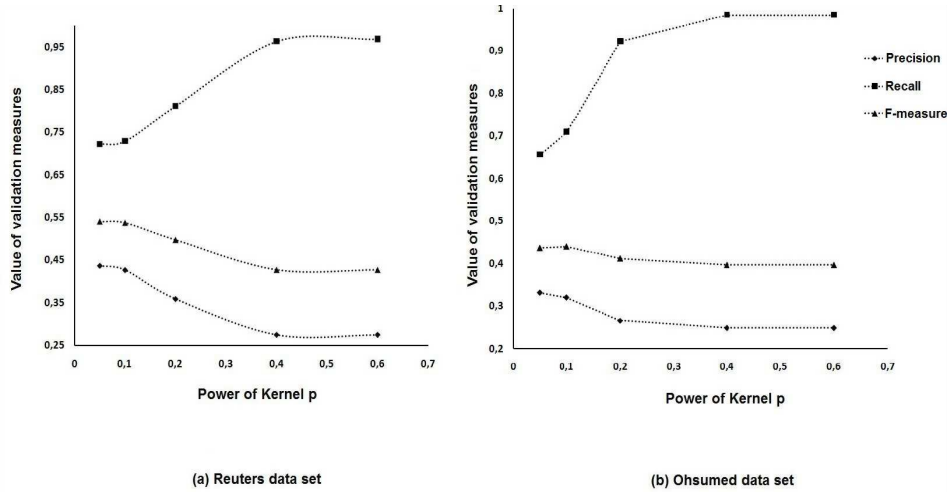


Fig. 5. The impact of varying power of sub polynomial kernel p on the performance of KOKM based WSK

data need to be assigned to more than one cluster and cannot be described by explicit feature vectors but described by strings such as the analysis of phylogenetic profiles. For such type of applications, it should be interesting to investigate the application of an overlapping clustering process based on sequences of text.

References

1. R. M. Aliguliyev,"Clustering of document collection a weighting approach," *Expert Systems with Applications*, **36:4** (2009) 7904–7916.
2. A. Banerjee, C. Krumpelman, S. Basu, R. J. Mooney, and J. Ghosh, "Model based overlapping clustering," *In International Conference on Knowledge Discovery and Data Mining*, Chicago, USA, 2005, pp 532–537.
3. N. Cancedda, E. Gaussier, C. Goutte, and J.M. Renders,"Word-sequence kernels," *Journal of Machine Learning Research*, **3** (2003) 1059–1082.
4. M. E. Celebi, H. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, **40** (2013) 200–210.
5. Y.C. Liu, C. Wu, and M. Liu, "Research of fast som clustering for text information," *Expert Systems with Applications*, **38:8**(2011) 9325–9333.
6. G. Cleuziou, "An extended version of the k-means method for overlapping clustering," *In International Conference on Pattern Recognition ICPR*, Florida, USA, 2008, pp 1–4.
7. C-E. Ben N'cir, G. Cleuziou, and N. Essoussi, "Overview of overlapping partitional clustering methods," *Partitional Clustering Algorithms*, **1** (2014) 245–276.
8. C-E. Ben N'cir and N. Essoussi, "Overlapping patterns recognition with linear and non-linear separations using positive definite kernels," *International Journal of Computer Applications*, **56:9**(2012) 1–8.
9. M-P. Fay and M-A. Proschan, "Wilcoxon-Mann-Whitney or t-test? On assumptions

for hypothesis tests and multiple interpretations of decision rules," *Statistics Surveys*, **4** (2010) 1–39.
10. K. Heller and Z. Ghahramani, "A nonparametric bayesian approach to modeling overlapping clusters," *Journal of Machine Learning Research*, **2** (2007) 187–194.
11. D. Isa, V-P. Kallimani and L-H. Lee, "Using the self organizing map for clustering of text documents," *Expert Systems with Applications*, **36:5** (2009) 9584 – 9591.
12. W. Jason, S. Bernhard, E. Eleazar, L. Christina, and N. William, "Dealing with large diagonals in kernel matrices," *Annals of the Institute of Statistical Mathematics*, **55:2** (2003) 391–408.
13. P. Lingras and C. West, "Interval set clustering of web users with rough k-means," *Journal of Intelligent Information Systems*, **23:1** (2004) 5–16.
14. H. Lodhi, N. Cristianini, J. Shawe-Taylor, and C. Watkins, "Text classication using string kernel," *The Journal of Machine Learning Research*, **2** (2001) 419–444.
15. H-B. Mann and D-R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, **18** (1947) 50–60.
16. G. Oberreuter and J-D. Veläsquez, "Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style," *Expert Systems with Applications*, **40:9** (2013) 3756 – 3763.
17. P. Pantel and D. Lin, "Discovering word senses from text," *In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Alberta, Canada, 2002, pp 613–619.
18. R. Saraçoglu and K. Tütüncü and N. Allahverdi, "A new approach on search for similar documents with multiple categories using fuzzy clustering," *Expert Systems with Applications*, **34:4** (2008) 2545–2554.
19. R. Saraçoglu and K. Tütüncüand N. Allahverdi, "A fuzzy clustering approach for finding similar documents using a novel similarity measure," *Expert Systems with Applications*, **33:3** (2007) 600 – 605.
20. C-G-M. Snoek, M. Worring, J. van Gemert, J-M. Geusebroek, and A-W-M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," *In Proceedings of the 14th annual ACM international conference on Multimedia*, New York, USA, 2006, pp 421–430.
21. G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," *In Data Mining and Knowledge Discovery Handbook*, chapter 34, 2010, pp 667–685.
22. A. Wieczorkowska, P. Synak, and Z. Ras, "Multi-label classification of emotions in music," *In Intelligent Information Processing and Web Mining*, volume 35 of *Advances in Soft Computing*, 2006, pp 307–315.
23. Y. Yang, "An evaluation of statistical approaches to text categorization," *Journal of Information Retrieval*, **1** (1999) 67–88.
24. E.J. Yannakoudakis, I. Tsomokos, and P.J. Hutton, "n-grams and their implication to natural language understanding," *Pattern Recognition*, **23:5** (1990) 509 – 528.
25. S. Zhang, R-S. Wang, and X-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A: Statistical Mechanics and its Applications*, **374:1** (2007) 483–490.

Biographical Sketch and Photo

Chiheb-eddine Ben N'Cir received the Ph.D. degree in computer engineering from Higher Institute of Management, University of Tunis, in 2014. Currently, he is an assistant professor at FSEGS, University of Sfax and a member of LARODEC laboratory. His research interests concern Unsupervised Learning methods and Data Mining tools with a special emphasis on non-disjoint partitioning of data.

Nadia Essoussi is an associate professor of computer science at the Higher Institute of Management, University of Tunis. She is a member of the LARODEC Laboratory and her research interests include Data Mining and Machine Learning with a particular focus on overlapping data and more recently Big Data Analytics.