Approximate Representation of Textual Documents in the Concept Space

Jasminka Dobša
University of Zagreb, Faculty of Organization and Informatics
Pavlinska 2, 42 000 Varaždin, Croatia
jasminka.dobsa@foi.hr

Bojana Dalbelo Bašić
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10 000 Zagreb, Croatia
Bojana.Dalbelo@fer.hr

Keywords: dimensionality reduction, concept decomposition, information retrieval

Received: November 17, 2006

In this paper we deal with the problem of addition of new documents in collection when documents are represented in lower dimensional space by concept indexing. Concept indexing (CI) is a method of feature construction that is relying on concept decomposition of term-document matrix. By using CI original representations of documents are projected on the space spread by centroids of clusters, which are called concept vectors. This problem is especially interesting for application on World Wide Web. Proposed methods are tested for the task of information retrieval.
Vectors on which the projection is done in the process of dimension reduction are constructed on the basis of representations of all documents in the collection, and computation of the new representations in the space of reduced dimension demands recomputation of concept decomposition. The solution to this problem is the development of methods which will give approximate representation of newly added documents in the space of reduced dimension.
In the paper are introduced two methods for addition of new documents in the space of reduced dimension. In the first method there no addition of new index terms and added documents are represented by existing list of index terms, while in the second method list of index terms is extended and representations of documents and concept vectors are extended in dimensions of newly added terms. It is shown that representation of documents by extended list of index terms does not improve performance of information retrieval significantly.

Povzetek: Predstavljeni sta dve metodi konceptualnega indeksiranja dokumentov.

1 Introduction

In this paper we deal with the problem of addition of new documents in collection when documents are represented in lower dimensional space by concept indexing. This problem is especially interesting for application on World Wide Web. Proposed methods are tested for the task of information retrieval [1].

There are lots of motives for dimension reduction in the vector space model: decrease of memory space needed for representation of documents, faster performance of information retrieval or automatic classification of documents, reduction of noise and redundancy present in the representation of documents. Methods for dimension reduction in the vector space model based on extraction of new parameters for representation of documents (feature construction) tend to overcome the problem of synonyms and polysemies which are two major obstacles in information retrieval. Disadvantage of feature construction may be uninterpretability of newly obtained parameters or features.

Our investigation is based on the method of feature construction called *concept indexing* which was introduced in 2001 by Dhillon and Modha [7]. This method uses centroids of clusters created by the spherical k-means algorithm or so-called *concept decomposition* (CD) for lowering the rank of the term-document matrix. By using CI original representations of documents are projected on the space spread by centroids of clusters, which we call here *concept vectors*.

Representation of new document in the vector space model is trivial. The problem appears when we want to add new documents in the space of reduced dimension. Namely, vectors on which the projection is done in the process of dimension reduction are constructed on the basis of representations of all documents in the collection, and computation of the new representations in the space of reduced dimension demands recomputation of the concept decomposition. The solution to this problem is the development of methods which will give approximate representation of newly added documents in

the space of reduced dimension. Application of such a methods will delay a process of recomputation of concept decomposition.

Methods for addition of representations of new documents in the space of reduced dimension are already developed for LSI method [3,9]. The method of LSI was introduced in 1990 [4] and improved in 1995 [3]. Since then LSI is a benchmark in the field of dimension reduction. Although the LSI method has empirical success, it suffers from the lack of interpretation of newly obtained features which causes the lack of control for accomplishing specific tasks in information retrieval. Kolda and O'Leary [8] developed a method for addition of representations of new documents for LSI method that uses semi-discrete decomposition which saves memory space.

When the collection of documents is extended it seems natural to extend also the list of index terms with terms present in added documents, which were not present in starting collection of documents, or were present very rarely and they were not included in the list of the index terms. In the paper are introduced two methods for addition of new documents in the space spread by concept vectors, which is called *concept space.* In the first method there no addition of new index terms and added documents are represented by existing list of index terms, while in the second method list of index terms is extended and representations of documents and concept vectors are extended in dimensions of newly added terms.

This paper is organized as follows. Section 2 provides a description of technique of dimensionality reduction by concept decomposition. In Section 3 novel algorithms for approximate addition of documents in concept space are proposed. Section 4 provides an example, while Section 5 describes experiment where proposed algorithms are tested. Last section gives conclusions and directions for further work.

2 Dimensionality reduction by the concept decomposition

Let the $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ be the term-document matrix. Then a_{ij} is the weight of the i-th term in the j-th document. A query has the same form as a document; it is a vector whose i-th component is the weight of the i-th term in the query. A common measure of similarity between the query and the document is the cosine of the angle between them.

Techniques of feature construction enable mapping documents' representations, which are similar in their content, or contain many index terms in common, to the new representations in the space of reduced dimension, which are closer than their representations in original vector space. That enables retrieving of documents which are relevant for the query, but do not contain index terms contained in the vector representation of query.

In this section we will describe the algorithm for computation of concept decomposition by the fuzzy k-mans algorithm [5].

2.1 Fuzzy k-means algorithm

The fuzzy k-means algorithm (FKM) [10] generalizes the hard k-means algorithm. The goal of the k-means algorithm is to cluster n objects (here documents) in k clusters and find k mean vectors or centroids for clusters. Here we will call these mean vectors *concept vectors*, because that is what they present. As opposed to the hard k-means algorithm, which allows a document to belong only to one cluster, FKM allows a document to partially belong to multiple clusters. FKM seeks a minimum of a heuristic global cost function $J_{fuzz} = \sum_{i=1}^{k}\sum_{j=1}^{n} \mu_{ij}^{b} \left\| \mathbf{a}_j - \mathbf{c}_i \right\|$ where $\mathbf{a}_j, j = 1,\ldots,n$ are vectors of documents, $\mathbf{c}_i, i = 1,\ldots,k$ are concept vectors, μ_{ij} is the fuzzy membership degree of document $\mathbf{a}_j$ in the cluster whose concept is $\mathbf{c}_i$ and b is a weight exponent of the fuzzy membership.

In general, the J_{fuzz} criterion is minimized when concept $\mathbf{c}_i$ is close to those documents that have a high fuzzy membership degree for cluster i , $i = 1,\ldots,k$. By solving a system of equations $\frac{\partial J_{fuzz}}{\partial \mathbf{c}_i}$ and $\frac{\partial J_{fuzz}}{\partial \mu_{ij}}$, we obtain a stationary point for which fuzzy membership degrees are given by

$$\mu_{ij} = \frac{1}{\sum_{r=1}^{k} \left(\frac{\left\| \mathbf{a}_j - \mathbf{c}_i \right\|^2}{\left\| \mathbf{a}_j - \mathbf{c}_r \right\|^2} \right)^{\frac{1}{b-1}}} \qquad (1)$$

for $i = 1,\ldots,k$ and $j = 1,\ldots,n$, while centroids or concept vectors are given by

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{b} \mathbf{a}_j}{\sum_{j=1}^{n} \mu_{ij}^{b}} \qquad (2)$$

for $i = 1,\ldots,k$. For such a stationary point the cost function reaches a local minimum. We will obtain concept vectors by starting with arbitrary initial concept vectors $\mathbf{c}_i^{(0)}, i = 1,\ldots,k$ and by computing fuzzy membership degrees $\mu_{ij}^{(t)}$, cost function $J_{fuzz}^{(t)}$ and new

concept vectors $\mathbf{c}_i^{(t+1)}$ iterative, where t is the index of iteration, until $\left|J_{fuzz}^{(t+1)} - J_{fuzz}^{(t)}\right| < \varepsilon$ for some threshold ε.

2.2 Concept decomposition

Our target is to approximate each document vector by a linear combination of concept vectors. The *concept matrix* is an $m \times k$ matrix whose j-th column is the concept vector $\mathbf{c}_j$, that is $\mathbf{C}_k = [\mathbf{c}_1, \ldots, \mathbf{c}_k]$. If we assume linear independence of the concept vectors, then it follows that the concept matrix has rank k. Now we define the *concept decomposition* $\mathbf{P}_k$ of the term-document matrix $\mathbf{A}$ as the least-squares approximation of $\mathbf{A}$ on the column space of the concept matrix $\mathbf{C}_k$. Concept decomposition is an $m \times n$ matrix $\mathbf{P}_k = \mathbf{C}_k \mathbf{Z}^*$ where $\mathbf{Z}^*$ is the solution of the least-squares problem, ie. $\mathbf{Z}^* = \left(\mathbf{C}_k^T \mathbf{C}_k\right)^{-1} \mathbf{C}_k^T \mathbf{A}$.

$\mathbf{Z}^*$ is a matrix of the type $k \times n$ and its columns are representations of documents in the concept space. Similarly, representation of query $\mathbf{q}$ in the reduced dimension space is given by $\left(\mathbf{C}_k^T \mathbf{C}_k\right)^{-1} \mathbf{C}_k^T \mathbf{q}$ and similarity between document and the query is given by the cosine of the angle between them. Concept indexing is a technique of indexing text documents by using concept decomposition.

3 Addition of representations of new documents in the concept space

In this section novel algorithms for addition text documents' representations in the concept space are proposed. The goal is to add new documents in a collection represented in the reduced dimension space, and this goal is achieved with and without an extension of the list of the index terms.

Let us introduce matrix notation that will be used in the section. Matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \tag{3}$$

will be an extended term-document matrix, where $\mathbf{A}_1$ a is matrix of starting documents in the space of starting terms, $\mathbf{A}_3$ is a matrix of starting documents in the space of added terms, $\mathbf{A}_2$ is a matrix of added documents in the space of starting terms and $\mathbf{A}_4$ is a matrix of added documents in the space of added terms. Further, let m_1 be number of starting terms, m_2 number of added terms, n_1 number of starting documents and n_2 number of added documents.

Here we will introduce two methods of approximate addition of new documents in the concept space:

(a) projection of new documents on existing concept vectors (Method A) and,
(b) projection of new documents on existing concept vectors extended in dimensions of newly added terms (Method B).

Assume that documents of a starting matrix $\mathbf{A}_1$ are clustered by fuzzy k-means algorithm and centroids of clusters are computed. Let $\mathbf{C}_1$ be the concept matrix the columns of which are concept vectors and let $\mathbf{C}_2$ be a matrix consisting of extensions of concept vectors in dimensions of added terms. Concept vectors of the matrix $\mathbf{C}_1$ are calculated by the formula (2) using columns of matrix $\mathbf{A}_1$ as document representations, while extensions of concept vectors are calculated by the same formula using respective columns of matrix $\mathbf{A}_3$ as representations of starting documents in the space of added terms. Let extensions of concept vectors form extension of the concept matrix denoted by $\mathbf{C}_2$. Then

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \tag{1}$$

is the concept matrix the columns of which are concept vectors extended in dimensions of newly added terms. Representations of documents in the concept space of extended term-document matrix will be given by expression

$$\begin{aligned}
&\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}\right)^{-1} \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= \left(\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2\right)^{-1} \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&\approx \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \begin{bmatrix} \mathbf{C}_1^T & \mathbf{C}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= \begin{bmatrix} \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T & \vdots & \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= [\left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T \mathbf{A}_1 + \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \mathbf{A}_3 \\
&\vdots \quad \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T \mathbf{A}_2 + \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \mathbf{A}_4] \\
&= [(5)+(6) \quad \vdots \quad (7)+(8)]
\end{aligned} \tag{4}$$

In the third line of the expression (4) it is assumed approximation $\left(\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2\right) \approx \mathbf{C}_1^T \mathbf{C}_1$. Such an approximation is justified by the fact that extensions of concept vectors are sparser than concept vectors formed from starting documents, because the coordinates of extended concept vectors are weights of added terms which were not included in list of the index terms before addition of new documents. It was established, by experiment, that $\left\|\mathbf{C}_2^T \mathbf{C}_2\right\|_2 << \left\|\mathbf{C}_1^T \mathbf{C}_1\right\|_2$. The number

of operations is significantly reduced by this approximation, because inverse $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}$ is already computed during the computation of starting documents projection.

This approximation is not necessary for the application of Method A, because this method does not use extensions of concept vectors. Representations of starting documents are given by expression (5), while representations of added documents are given by expression (7). Pre-processing of extended term-document matrix includes normalization of columns of matrices $\mathbf{A}_1$ (starting documents) and $\mathbf{A}_2$ (added documents) to the unit length. Let us now calculate number of operations needed for application of Method A. Representations of starting documents are already known, and so is matrix $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T$. That is why the number of operations is equivalent to the number of operations needed for multiplication of matrices $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T$ and $\mathbf{A}_2$, which is $2m_1kn_2$.

By the Method B added documents are projected on the space of extended concept vectors. Vector representations of starting documents are already known, and they are given by the (5), while representations of added documents are computed by the formula

$$\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T\mathbf{A}_2 + \alpha\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_2^T\mathbf{A}_4, \quad (9)$$

where coefficient $\alpha > 1$ has a role of stressing the importance of added terms and documents. Pre-processing of extended term-document matrix includes normalization of its columns to the unit length. Performance of Method B demands computation of concept vectors' extensions and computation of added documents projections. Computation of the first summand in formula (9) demands $2m_1kn_2$ operations, while computation of the second summand demands $(2k^2m_2+2m_2n_2k)$ operations, because inverse $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}$ is already calculated. Addition of matrix elements of the first and second summand and multiplication by scalar in the formula (10) demands $2n_2k$ operations. Further, computation of concept vectors extensions by application of formula (2) demands $(2n_1km_2+2n_1k)$ operations. Normalization of columns of extended term-document matrix and concept matrix is not included in calculation of number of operations, because it is a standard operation of pre-processing included in every algorithm. That means that application of Method B demands

$$\begin{aligned} N_B &= 2m_1kn_2 + 2k^2m_1 + 2m_2n_2k + 2n_1km_2 + 2n_1k + 2n_2k \\ &= 2k(m_1n_2 + km_2 + m_2n_2 + n_1m_2 + n_1 + n_2) \end{aligned}$$

operations.

4 An example

By this example [6] it will be shown, in an illustrative way, how documents are projected by CI method into the two-dimensional concept space. The collection of 19 documents (titles of books) will be used where 15 documents will form collection of starting documents and 4 documents will form the collection of added documents. The documents are categorized in three categories: documents from the field of data mining (DM documents), documents from the field of linear algebra (LA documents) and documents which combine these two fields (application of linear algebra on data mining). The documents with their categorization are listed in Table 1. A list of terms is formed from words contained in at least two documents of starting collection, after which words on the stop list are ejected and variations of words are mapped on the same characteristic form (e.g. the terms *matrix* and *matrices* are mapped on the term *matrix*, or *applications* and *applied* are mapped on *application*). As a result, a list of 16 terms is obtained which we have divided in three parts: 8 terms from the field of data mining (*text, mining, clustering, classification, retrieval, information, document, data***)**, 5 terms from the field of linear algebra (*linear, algebra, matrix, vector, space***)** and 3 neutral terms (*analysis, application, algorithm***)**. Then we have created a term-document matrix from starting collection of documents and normalized the columns of it to be of the unit norm. This is a term-document matrix of starting documents in the space of starting terms $\mathbf{A}_1$. Then we have applied CD (k=2) to that matrix. In CD $\mathbf{C}_2\mathbf{Z}^*$ rows of concept matrix $\mathbf{C}_2$ are representations of terms and columns of $\mathbf{Z}^*$ are representations of documents of starting collection.

We have also created two queries (underlined words are from the list of terms):

1) Q1: Data mining
2) Q2: Using linear algebra for data mining.

For Q1 all data mining documents are relevant, while for Q2 documents D6, D18 and D19 are relevant. Most of the DM documents do not contain words *data* and *mining***.** Such documents will not be recognized by the simple term-matching vector space method as relevant. Documents D6 and D19, which are relevant for Q2, does not contain any of terms from the list contained in the query. The representation of the query $\mathbf{q}$ by concept indexing will be $\tilde{\mathbf{q}} = (\mathbf{C}_k^T\mathbf{C}_k)^{-1}\mathbf{C}_k^T\mathbf{q}$ and in the same way will be computed representations of added documents' collection (application of Method A).

Number	Status (Starting/Added)	Categorization	Document
D1	Starting	DM	Survey of text mining: clustering, classification, and retrieval
D2	Starting	DM	Automatic text processing: the transformation analysis and retrieval of information by computer
D3	Starting	LA	Elementary linear algebra: A matrix approach
D4	Starting	LA	Matrix algebra and its applications in statistics and econometrics
D5	Starting	DM	Effective databases for text and document management
D6	Starting	Combination	Matrices, vector spaces, and information retrieval
D7	Starting	LA	Matrix analysis and applied linear algebra
D8	Starting	LA	Topological vector spaces and algebras
D9	Starting	DM	Information retrieval: data structures and algorithms
D10	Starting	LA	Vector spaces and algebras for chemistry and physics
D11	Starting	DM	Classification, clustering and data analysis
D12	Starting	DM	Clustering of large data sets
D13	Starting	DM	Clustering algorithms
D14	Starting	DM	Document warehousing and text mining: techniques for improving business operations, marketing and sales
D15	Starting	DM	Data mining and knowledge discovery
D16	Added	DM	Concept decomposition of large sparse text data using clustering
D17	Added	LA	A rank-one reduction formula and its applications to matrix factorizations
D18	Added	Combination	Analysis of data matrices
D19	Added	Combination	A semi-discrete matrix decomposition for latent semantic indexing in information retrieval

Table 1: Documents and their categorization (DM – data mining documents, LA – linear algebra documents). Documents D6, D18 and D19 are combination of these two categories. Words from the list of terms are underlined.

In Figure 1 are shown images of representations of documents and queries in the concept space. It can be seen that LA documents of starting collection are grouped (and located near *x* axes); DM documents of starting collection are somewhat more dispersed, but generally also grouped around *y* axes, while D6 document (combination) is in the group of LA documents. It appears that way because during the clustering by fuzzy k-means algorithm D6 document was clustered to group of LA documents. Namely, fuzzy k-means algorithm allows documents to belong to multiple clusters partially during the process of clustering, but the result of convergence are hard partitions, which means that at the end algorithm decide in which cluster document belong.

Shaded areas on Figure 1 represent the areas of relevant documents for queries in the cosine similarity sense (cosine of the angle between points in shaded areas and representation of the queries is greater than 0.9). The added documents are shown on the figure in the shape that correspond to the category they belong, but in lighter colour then documents of the starting collection. By usage of the Method A document D16 (DM document) is

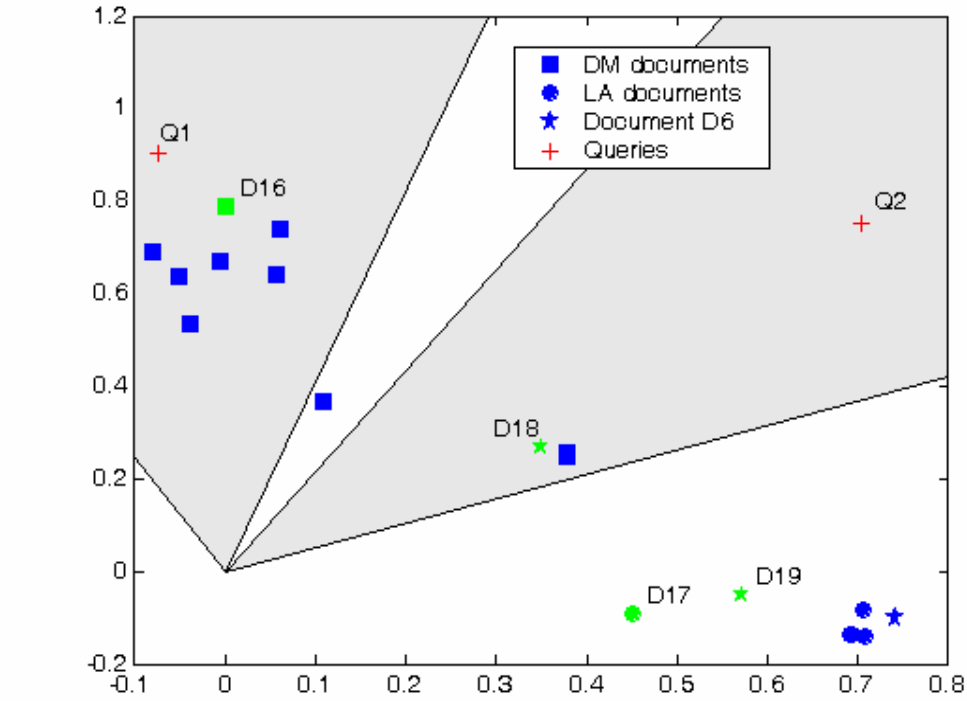

Figure 1: Representations of starting and added documents in the concept space. Representations of added document are shown in the shape that correspond to category of document, but in lighter colour then representations of starting documents. Shaded areas are areas of relevant documents for queries.

mapped in the group of DM documents and D17 document (LA document) is mapped near group of LA documents. Document D19 which combines fields of linear algebra and data mining is mapped near LA documents (because it is represented by index terms similarly as document D6) and document D18 which contains index term *data* also contained in the query Q2 is mapped in the area of relevant documents for Q2 query.

5 Experiment

Experiments are conducted on MEDLINE collection of documents. The collection contains 1033 documents (abstracts of medical scientific papers) and 35 queries. The documents of collection are split randomly into two parts: starting documents and added documents. The ratio of starting and added documents is varied: first added documents form 10% of the whole collection, then 20% of the whole collection, and so on. Starting list of index terms is formed on the basis of starting collection of documents. In the list are included all words contained in at least two documents of starting collection, which are not on the list of stop words. Further, the list of index terms is formed for the whole collection of documents in an analogous way. The obtained list of index terms for the whole collection contains 5940 index terms.

We have used measure of mean average precision (MAP) [1] for evaluation of the experimental results. Concept decomposition is conducted under starting collection of documents and added documents are represented in the concept space by using one of the described methods for approximate addition of documents. After that, an evaluation of information retrieval performance is conducted under the whole collection of documents. Dimension of the space of reduced dimension is fixed to $k=75$.

In the first row of Table 2, there is MAP of information retrieval in the case that procedure of concept decomposition is conducted under whole collection of documents (percentage of added documents is 0%). This value presents MAP in the case of recomputation of concept decomposition when new documents are added in the collection. All other values of MAP in the cases when the collection is divided into collection of starting and added documents in the different ratios, could be compared to this value. The second column of Table 2 presents number of added documents, while the third column presents number of added terms. Let us note that number of added terms grows linearly, and that the collection with only 20% of starting documents is indexed with a much smaller set of index terms then the whole collection. The fourth row presents MAP for approximate addition of documents by Method A .

Percentage of added documents	**Number of added documents**	**Number of added terms**	**MAP** Method A	**MAP** Method B α=1.0	**MAP** Method B α=1.5	**MAP** Method B α=2.0
0	0	0	54.99	54.99	54.99	54.99
10	104	456	51.98	52.20	52.33	**52.37**
20	208	753	54.96	55.10	55.09	**55.23**
30	311	1264	51.90	51.78	51.97	**52.03**
40	414	1673	50.84	50.60	51.09	**51.64**
50	517	2089	**48.64**	47.99	48.29	**48.64**
60	620	2696	44.26	44.08	45.04	**45.49**
70	723	3282	**43.59**	41.86	42.32	42.70
80	826	4024	39.87	40.56	42.56	**43.74**

Table 2: Mean average precision of information retrieval for approximate addition of new documents by Method A (without addition of new index terms) and Method B (with addition of new index terms) compared for different splits of document collection. Parameter α (used in Method B) has a role of additional stressing the importance of added terms and documents. The best results for every split of document collection are shown bolded. Generally, the best results are achieved for Method B, α=2.0, but these results are not significantly better in comparison to results obtained by Method A.

The rest columns of Table 2 present MAP of information retrieval for approximate addition of new documents by Method B for different values of parameter α.

The best results for every split of documents are show bolded. From the results we can conclude that an addition of new index terms does not improve results of MAP significantly. Namely, results obtained by Method B are better then results achieved by Method A and additional stressing of added terms and documents (for α>1) has positive effect on results. Nevertheless, results obtained by Method B, α=2.0 are not significantly better in comparison to results obtained by Method A according to pared t-test (α=0.05).

6 Conclusions and future work

Values of MAP for approximate methods are acceptable in comparison to repeated computation on concept decomposition when the number of added documents is the same or smaller than the number of starting documents. There is a drop of MAP when the number of added documents exceeds the number of starting documents. Results of MAP are not significantly improved by the methods that use extended list of index terms obtained as a result of addition of documents. It is interesting to notice that this statement is valid even in the cases when the list of index terms is significantly enlarged, which is when larger proportion of documents is added. This results show a great redundancy present in the textual documents.

In the future we plan to develop new methods of approximate addition of documents that will correct existing concept vectors by using the representations of added documents.

References

[1] R. Baeza-Yates, B.Ribeiro-Neto. *Modern Information Retrieval*, Addison-Wesley, ACM Press, New York, 1999.

[2] M. W. Berry, Z. Drmač, E. R. Jessup. Matrices, Vector Spaces, and Information Retrieval, *SIAM Review*, Vol. 41. No. 2, 1999, pp. 335-362.

[3] M. W. Berry, S. T. Dumais, G. W. O'Brien. Using linear algebra for intelligent information retrieval, *SIAM Rewiew,* Vol. 37. 1995, pp. 573-595.

[4] S. Deerwester, S. Dumas. G. Furnas. T. Landauer, R. Harsman. Indexing by latent semantic analysis, *J. American Society for Information Science,* Vol. 41. 1990, pp. 391-407.

[5] J. Dobša, B. Dalbelo-Bašić. Concept decomposition by fuzzy k-means algorithm, *Proceedings of the IEEE/WIC International Conference on Web Intelligence, WI 2003,* 2003, pp. 684-688.

[6] J. Dobša, B. Dalbelo-Bašić, Comparison of information retrieval techniques: latent semantic indexing and concept indexing, *Journal of Inf. and Organizational Sciences*, Vol.28 , No. 1-2, 2004, pp.1-17

[7] I. S. Dhillon, D. S. Modha, Concept Decomposition for Large Sparse Text Data using Clustering, *Machine Learning* , Vol. 42. No. 1, 2001, pp. 143-175.

[8] T. Kolda, D. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing in information retrieval, *ACM Trans. Inform. Systems,* Vol. 16, 1998, pp. 322-346.

[9] G.W. O'Brien. *In formation Management Tools for Updating an SVD-Encoded Indexing Scheme,* Master s thesis, The University of Knoxville, Tennessee, 1994.

[10] J. Yen. R. Langari. *Fuzzy Logic: Intelligence, Control and Information,* Prantice Hall, New Jersey, 1999.