Composition of Methods in Document Clustering

Kristóf Csorba *
kristof@aut.bme.hu

István Vajk*
vajk@aut.bme.hu

Abstract: Document clustering is a widely researched area of information retrieval. As there are many possibilities for noise filtering and other performance improving tricks, this paper focuses on the comparison of some techniques in latent semantic indexing, which aims to track semantically related terms to decrease feature space dimensionality. 5 tile methods (term filtering, frequency quantizing, principal component analysis (PCA), term clustering and document clustering of course) are used to build various configurations. These are then compared based on maximal achieved F-measure and time consumption to find the best composition.

Keywords: Latent semantic indexing, singular value decomposition, double clustering, term frequency quantizing

1 Introduction

Document clustering is a procedure to separate documents according to certain criteria, for instance their topics. As there isn't an exact definition for topic, there are many ways to identify it. A common approach is based on common terms in the documents: two documents are said to be of similar topic if they share many terms. (As even humans often disagree how to order a set of documents into two groups, a globally optimal solution is said not to exist.)

Vector space models [1] create a feature space to characterize the documents and comparisons are performed after a transformation into this space. The first step is usually the creation of the document–term frequency (or occurrence number) matrix, which contains the number of occurrences of a term (represented by a row) in a document (represented by a column). This matrix is created as a result of parsing of the documents.

One could say that as the (unsupervised) clustering of the documents is equivalent to the clustering of the column vectors of the document–term matrix, a clustering method could be applied directly to this matrix to get the solution. The most important problem is caused by the terms with similar meaning: even documents with very similar topic tend to contain different words (like synonyms), which makes their column vectors too far from each other.

2 Tiles for document clustering

The process of unsupervised document clustering starts in this case by generating the n-by-m document–term matrix $\mathbf{X}$. The element $\mathbf{X}_{i,j}$ is the occurrence number of the i-th term in the j-th document.

In the following the possible tile methods used in the experiments are described in detail.

* Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Goldmann Gy. tr 3, Budapest, Hungary, H-1111

2.1 Term filters

Term filtering is an essential way to reduce the dimensionality of the feature space. Stopword removal is a very common step, which removes words like "the", "a", "with", etc. from the documents, as these cannot be used to identify any topics. In this case, after stopword removal a term frequency based filtering was applied to all the documents. (Other possibilities would be for instance techniques based on information gain [2] or part of speech tagging.)

Given the document–term occurrence matrix $\mathbf{X}$, using a norm defined by the sum of occurrences

$$\|\mathbf{v}\| = \sum_{i=1}^{n} \mathbf{v}_i \tag{1}$$

the term filter removes terms (row vectors of $\mathbf{X}$) i for which

$$\|(\mathbf{X})_{i,\bullet}\| < MinFrequency \tag{2}$$

The remaining rows build the $\mathbf{X}$ used in the ongoing processes. As this filtering step is very important due to time consumption reasons, all the tested configurations in our experiments contain this tile as the first step.

2.2 Quantizing

Considering the nature of terms the exact number of occurrences of a given term in a document is relative unimportant from the topic's point of view. One can define frequency categories like "never", "rare", "frequent", but a deeper level of granularity makes only the comparison of the documents harder.

Based on this assumption a frequency quantizing filter can be employed to remove unnecessary diversity. The quantizing procedure assigns new values to the elements of $\mathbf{X}$ based on a quantizing list $\mathbf{q}$, which contains the possible quantized values, like $[0, 1, 2]$ for example.

$$quantize(x) = \max_i \{\mathbf{q}_i \leq x\} \tag{3}$$

$$\mathbf{X}_{i,j} = quantize((\mathbf{X}_f)_{i,j}) \tag{4}$$

2.3 Principal component analysis

Singular value decomposition [3] is a matrix-analytical method to search for a space, where a given linear transformation (usually given by a rectangular matrix) is a scaling along the base vectors. More precisely if $\mathbf{X}$ is an n-by-m matrix, SVD has a result in the form

$$\mathbf{X} = \mathbf{USV}^T \tag{5}$$

where $\mathbf{S}$ is the r-by-r diagonal matrix of the singular values of $\mathbf{X}$ (r is the rank of $\mathbf{X}$) and $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns. [1]

If the row vectors of $\mathbf{U}$ and $\mathbf{V}$ are taken geometrically as coordinates in the m dimensional space, the terms and the documents became points in a vector space. In this "feature space" the coordinates are linear combinations of documents (or terms respectively). The diagonal of

[1] SVD of $\mathbf{X}$ is very close to the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$ into the form $\mathbf{VLV}^T$, where $\mathbf{V}$ contains the eigenvectors in it's columns and $\mathbf{L}$ the corresponding eigenvalues. In fact, $\mathbf{U}$ consists of the eigenvectors of $\mathbf{XX}^T$ and $\mathbf{V}$ of the eigenvectors of $\mathbf{X}^T\mathbf{X}$. $\mathbf{L}$ is equivalent to $\mathbf{S}^2$ in this case.

$\mathbf{S}$ serves as a scaling of the axes in this space reflecting the contribution of the dimensions in the overall similarity structure. As the singular values in the diagonal of $\mathbf{S}$ are provided in descending order, one can easily select the most significant components.

The PCA step can be used to generate a reduced dimensionality representation of both the terms and the documents. For a compressed term representation $\mathbf{T}$ can be retrieved as follows:

$$\mathbf{T} = \mathbf{U}_{\bullet,1..kt}\mathbf{S}_{1..kt,1..kt} \tag{6}$$

In the feature space terms with similar occurrence behaviors are located near each other while words related to different topics (and used in different documents) are farer in the sense of cosine distance[2]. (A similar compressed representation of the documents can be retrieved as well from the $\mathbf{V}$ matrix.) In the next step, $\mathbf{T}$ can be used as a basis of the term clustering instead of the row vectors of $\mathbf{X}$.

2.4 Term clustering

Term clustering (the key of double clustering [4]) means the clustering of the term before document clustering to enable the clustering of the documents based on a document–term-cluster matrix instead of the document–term matrix. Terms are assigned to clusters and row vectors of the document–term matrix belonging to the same term-cluster are merged together.

Term clustering can be performed based on the $\mathbf{T}$ matrix retrieved from a previous PCA step or directly based on $\mathbf{X}$. The process based on the former way is as follows:

$$j_i = \alpha(\mathbf{T}_{i,\bullet}) \tag{7}$$

$$\alpha^{-1}(j) = \{\mathbf{T}_{i,\bullet} | \alpha(\mathbf{T}_{i,\bullet}) = j\} \tag{8}$$

The function $\alpha(\mathbf{T}_{i,\bullet})$ returns the index of the cluster the given term (represented by $\mathbf{T}_{i,\bullet}$) is assigned to. $\alpha^{-1}(j)$ returns the set of terms assigned to the $j-th$ term cluster. In our experiments the k-means algorithm was applied as the α function. The merging of the terms in a cluster is the simple addition of their representing row vectors:

$$\mathbf{Y}_{j,\bullet} = \sum_{\mathbf{v} \in \alpha^{-1}(j)} \mathbf{v} \tag{9}$$

$\mathbf{Y}$ is the document–term-cluster matrix containing term-clusters instead of terms as row vectors. The documents are represented as column vectors of $\mathbf{Y}$ as $\mathbf{D}_i = \mathbf{Y}^T_{\bullet,i}$. $\mathbf{Y}$ can be a basis of the document clustering in the next step instead of the $\mathbf{X}$ document–term matrix.

2.5 Document clustering

Document clustering is similar to the term clustering as follows:

$$j_i = \beta(\mathbf{T}_{i,\bullet}) \tag{10}$$

$$\beta^{-1}(j) = \{\mathbf{T}_{i,\bullet} | \beta(\mathbf{T}_{i,\bullet}) = j\} \tag{11}$$

As this step is essential to our goals, this tile method is used in every tested configurations, just like the term filtering mentioned earlier. In our experiments the β function is the k-means algorithm just as in the case of the term clustering.

[2] The cosine distance of two vectors $\mathbf{A}$ and $\mathbf{B}$ is the cosine of their angle: $\cos\varphi = \frac{\mathbf{AB}}{|\mathbf{A}||\mathbf{B}|}$.

3 The best configuration

To measure the capabilities of the tiles a pipe architecture (Fig. 1) was built, where every tile can be activated or deactivated independently. (Pipe architecture means in this case the execution of the tile elements after each other by connecting their outputs to the input of the next element.) The order of the tiles cannot be changed, because their functionality was designed with assumptions on the prior transformations. (For this reason, every tile has a "bypass" procedure, which performs a format transformation on the data to create a valid input for the next step without doing any theoretical important transformation.)

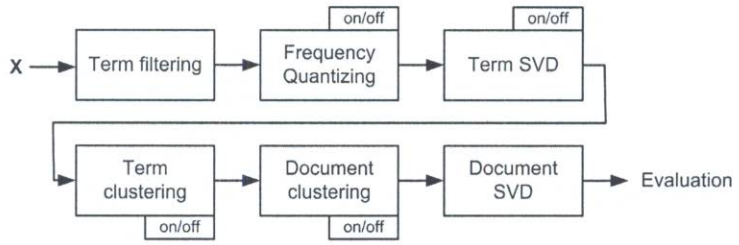

Fig. 1: Pipe of processing elements

As the testing data set, a part of the 20Newsgroups corpus ([5]) was selected: 250 documents from the "auto" and "graphics" section each. To find the best configuration, the behavior of the configurations were simulated and compared based on the performance measure F-measure [6]. The method comparison results are shown in Fig. 2. "Q" stands for the activated quantizing filter, "tSVD" for the term dimensionality reduction with SVD, "tClust" for term clustering and "dSVD" for the document dimensionality reduction (with SVD). The maximal achieved F-measure is taken over a wide range of parameter settings examined.

Based on the measurements the configuration resulting in the best performance contained frequency quantizing, term SVD and term clustering. (Document SVD was not employed in this case.)

The second most important aspect of performance is the time consumption. Fig. 3 shows, that the best configuration above is relative good in this aspect as well. Configurations with less active components are faster of course, but the little additional time is paying.

In general the followings can be stated:

- The quantizing filter is essential for F-measures over 0.9. It is active in every configuration with good clustering capabilities. This means the assumption, that the exact number of occurrences is not important, seems to be correct. The additional time need caused by this step seems not to be relevant.
- The term clustering (double clustering) is important for the good clustering performance as well, but its most important rule is to decrease the time consumption: all the fastest configurations contain term clustering. (Good clustering capability can be achieved without it as well.)
- "Term SVD" showed to be necessary for the best performance in F-measure and causes minimal additional time need when used with term clustering before it.
- "Document SVD" did not perform very well and seemed to decrease performance.

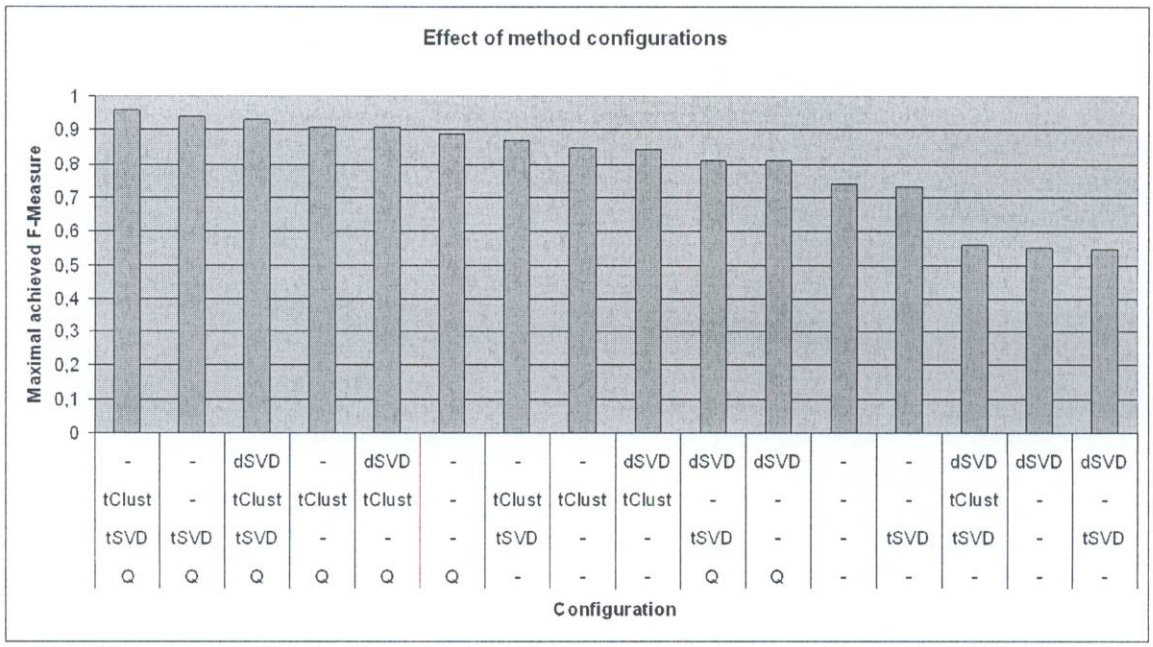

Fig. 2: Comparison of configurations, ordered by descending maximal F-measure

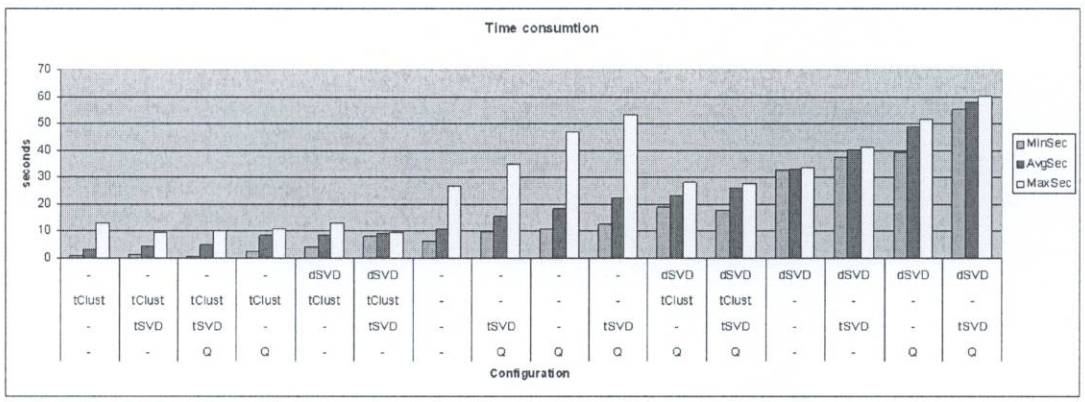

Fig. 3: Time consumption of configurations, ordered by average execution time

Used in its own (with only term filtering and document clustering together) the quantizing filter and the term clustering were able to overcome 0.8 F-measure, but none of them was enough to reach 0.9.

It should be noted, that all these results are valid if the parameters of the methods are set correctly. Bad parameter values can ruin even the best configurations. Examining the best configuration the best setting of the most important parameters are analyzed in the following section.

4 Important parameters

Taking the best configuration into account (quantizing, term SVD and term clustering) the followings are the most important parameters:

- q is the list of valid values after quantizing the term occurrence numbers.
- kt is the new dimension of the term feature vectors produced by term SVD.
- t is the number of term–clusters created by the term clustering.

The t number of term clusters has to be set intuitively to a value, which enables the separation of enough subtopics to distinguish between the terms of the document classes (standing for the two topics "auto" and "graphics"). This value usually moves between 5 and 12 clusters. The current situation is presented in Fig. 4. The best value is $t = 6$, but $t = 9$ was relative good as well.

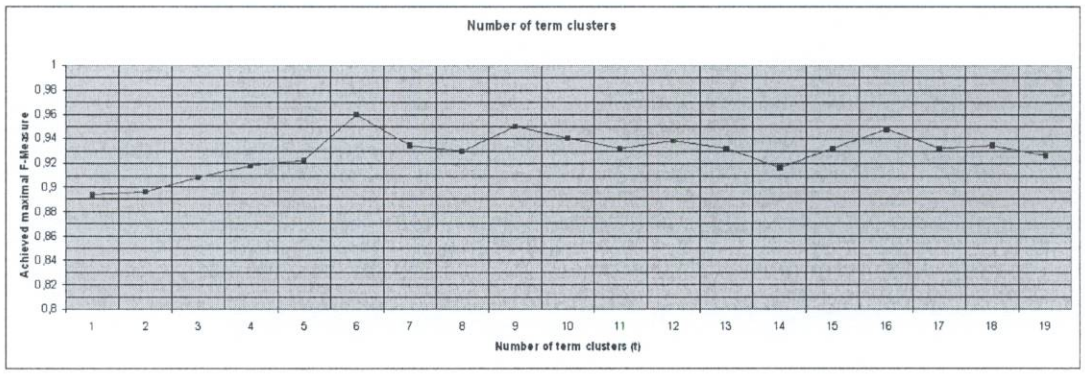

Fig. 4: Number of term clusters

The results with different kt values (term feature space dimensionality) can be seen in Fig. 5. As after the PCA every component tends to separate the terms into two groups, the optimal number of dimensions could be approximated as $kt_{opt} = \lceil log_2(t) \rceil$, because this is the number of dimensions needed to distinguish between t term-clusters. The results confirm this assumption as well. Too few dimensions cause term-clusters get too narrow to each other and leads to worse results.

The quantizing filter needs a list of allowed values. In Fig. 6 lists with one non-zero element are examined. (Zero values are always allowed to stand for terms not present in the document.) The best setting is clearly the list $[0; 1]$ which means a binary document–term occurrence matrix with values 0 and 1.

If we examine quantizing lists with two non-zero elements, performance is shown in Fig. 7. None of the settings showed an ability to overcome the performance of the $[0; 1]$ list. This is an interesting result as it states the uselessness of more than one non-zero quantizing value.

5 Conclusions

Based on the configuration selected above our system could reach 0.96 F-measure. This value is very promising, although it should be noted that the testing document set consisted only of two different topics and these two were relative far from each other.

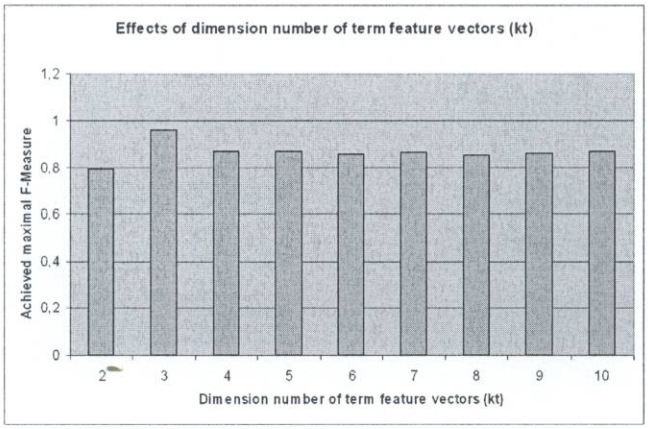

Fig. 5: Dimension number of term feature space

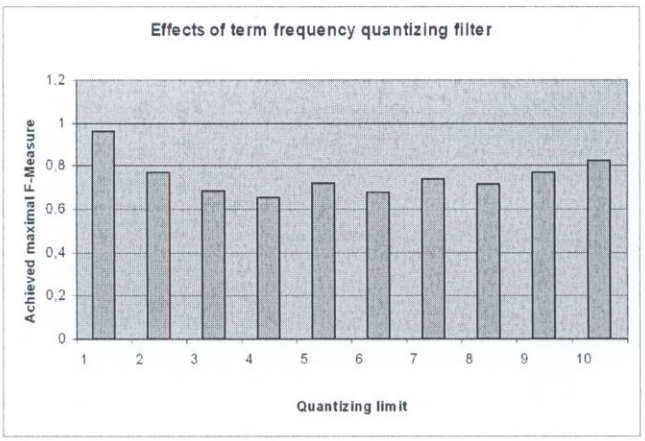

Fig. 6: Quantizing vectors with one non-zero elements

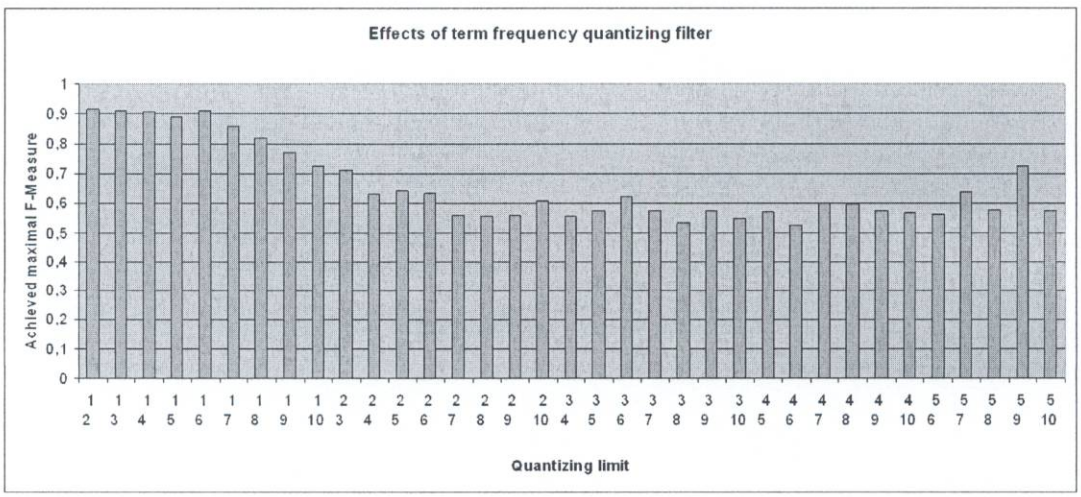

Fig. 7: Quantizing vectors with one non-zero elements

According to the components of the architecture the following conclusions can be made:

- The various settings of the quantizing filter were thought to bring more interesting results, but finally the pure binary quantizing was the best.
- The singular value decomposition was already known to be effective. The collective performance with the other components was shown here to be even better in the term–clustering phase, but not in the feature space of the documents: in that place SVD could not improve the performance.
- The term clustering was a very effective component, as it provides not only strong dimensionality reduction and acceleration, but also an effective noise reduction against words with similar meaning.
- The applicability of the system is limited primarily by the singular value decomposition as the most time consuming step if there are much more documents in the data set.

As the parameter settings influence the result in a very strong way, the selection of the optimal settings is still an open question. Some assumptions were made in this paper but this is still a subject of further research.

Bibliography

1. Singhal, A. (2001). *Modern information retrieval: A brief overview.* IEEE Data Engineering Bulletin, 24(4):35-43.
2. Li, L. and Chou, W. (2002). *Improving latent semantic indexing based classifier with information gain.* Technical report.
3. Furnas, G., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R., Streeter, L. A., and Lochbaum, K. E. (1988). *Information retrieval using a singular value decomposition model of latent semantic structure.* In Chiaramella, Y., editor, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 465-480, Grenoble, France. ACM.
4. Slonim, N. and Tishby, N. (2000). *Document clustering using word clusters via the information bottleneck method.* In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Clustering, pages 208-215.
5. Lang, K. (1995). *Newsweeder: Learning to filter netnews.* In ICML, pages 331-339.
6. Stein, B. and zu Eissen, S. M. (2003). *Automatic document categorization: Interpreting the perfomance of clustering algorithms.* In KI, pages 254-266.
7. Cristianini, N., Shawe-Taylor, J., and Lodhi, H. (2002). *Latent semantic kernels.* Journal of Intelligent Information Systems, 18(2/3):127-152. Special Issue on Automated Text Categorization.
8. Gong, Y. and Liu, X. (2001). *Generic text summarization using relevance measure and latent semantic analysis.* In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 19-25.
9. Liu, B., Chin, C. W., and Ng, H. T. (2003). *Mining topic-specific concepts and definitions on the web.* In WWW, pages 251-260.
10. Peshkin, L. and Pfeffer, A. (2003). *Bayesian information extraction network.* Intl. Joint Conference on Artificial Intelligence, 2003.