Improving Quality of Search Results Clustering with Approximate Matrix Factorisations

Stanislaw Osinski

Poznan Supercomputing and Networking Center,
ul. Noskowskiego 10, 61-704, Poznan, Poland
stanislaw.osinski@man.poznan.pl

Abstract. In this paper we show how approximate matrix factorisations can be used to organise document summaries returned by a search engine into meaningful thematic categories. We compare four different factorisations (SVD, NMF, LNMF and K-Means/Concept Decomposition) with respect to topic separation capability, outlier detection and label quality. We also compare our approach with two other clustering algorithms: Suffix Tree Clustering (STC) and Tolerance Rough Set Clustering (TRC). For our experiments we use the standard merge-then-cluster approach based on the Open Directory Project web catalogue as a source of human-clustered document summaries.

1 Introduction

Internet search engines have become an indispensable tool for people looking for information on the web. The majority of publicly available search engines adopt the so-called *query-list paradigm*, whereby in response to a user's query the search engine returns a linear list of short document summaries (*snippets*).

Despite its great popularity, the query-list approach has several deficiencies. If a query is too general, without a clear summary of different topics covered by the results, the users may have to go through a large number of irrelevant documents in order to identify the ones they were looking for. Moreover, especially in case of ill-defined queries, small groups of interesting but low-ranked outlier documents may remain unnoticed by most users.

One alternative to ranked lists is *search results clustering*. In this setting, in response to a query "london", for example, the user would be presented with search results divided into such topics as "London Hotels", "Weather Forecasts", "Olympic Games" or "London Ontario Canada". Users looking for information on a particular subject would be able to identify the documents of interest much quicker, while those who need a general overview of all related topics would get a concise summary of each of them.

Search results clustering involves a class of algorithms called post-retrieval document clustering algorithms [1]. A successful search results clustering algorithm must first of all identify the major and outlier topics dealt with in the results based only on the short document *snippets* returned by the search engine (most users are unwilling to wait for the full documents to download). Secondly,

M. Lalmas et al. (Eds.): ECIR 2006, LNCS 3936, pp. 167–178, 2006.

in order to help the users to identify the results of interest more quickly, the algorithm must label the clusters in a meaningful, concise and unambiguous way. Finally, the clustering algorithm must group the results fully automatically and must not introduce a noticeable delay to the query processing.

Many approaches to search results clustering have been proposed, including Suffix Tree Clustering (STC) [2], Semantic On-line Hierarchical Clustering (SHOC) [3], Tolerance Rough Set Clustering (TRC) [4], and DisCover [5]. With their respective advantages such as speed and scalability, all these algorithms share one important shortcoming: none of them explicitly addresses the problem of cluster description quality. This, unfortunately, leads these algorithms to *knowing* that certain documents should form a group and at the same time being unable to concisely *explain* to the user what the group's documents have in common.

Based on our previous experiences with search results clustering [6], we proposed an algorithm called Lingo [7] in which special emphasis was placed on the quality of cluster labels. The main idea behind the algorithm was to *reverse* the usual order of the clustering process: Lingo first identified meaningful cluster labels using the Singular Value Decomposition (SVD) factorisation, and only then assigned documents to these labels to form proper clusters. For this reason this algorithm could be considered as an example of a *description-comes-first* approach. Although SVD performed fairly well as part of Lingo in our experiments [8], it had certain limitations in the context of the description-comes-first approach. For this reason, we sought to verify how alternative matrix factorisations, known from e.g. image processing, would perform in place of SVD.

The aim of this paper is to compare how different matrix factorisations perform as parts of a description-comes-first search results clustering algorithm. We compare the factorisations with respect to major topic identification capability, outlier detection and cluster labels quality. We evaluate four factorisation algorithms: Singular Value Decomposition (SVD), Non-negative Matrix factorisation (NMF) [9], Local Non-negative Matrix Factorisation (LNMF) [10] and Concept Decomposition (CD) [11]. To further verify the viability the description-comes-first approach, we compare Lingo with two other algorithms designed specifically for clustering of search results: Suffix Tree Clustering (STC) and Tolerance Rough Set Clustering (TRC). We perform our experiments using data drawn from a large human-edited directory of web page summaries called Open Directory Project[1].

2 Related Work

The idea of search results clustering was first introduced in the Scatter/Gather system [12], which was based on a variant of the classic K-Means algorithm. Scatter/Gather was followed by Suffix Tree Clustering (STC) [13], in which snippets sharing the same sequence of words were grouped together. The Semantic

[1] http://dmoz.org

On-Line Hierarchical Clustering (SHOC) [3] algorithm used Singular Value Decomposition to group search results in the Chinese language according to the latent semantic relationships between the snippets. Yet another algorithm called DisCover [5] clustered search results in such a way as to maximise the coverage and distinctiveness of the clusters. Finally, there exist algorithms that use matrix factorisation techniques, such as Non-negative Matrix Factorisation [14], for clustering full text documents.

3 Background Information

3.1 Lingo: Description-Comes-First Clustering

In this section we provide a brief description of the Lingo algorithm, placing emphasis on its relation to matrix factorisation. For an in-depth formalised description and an illustrative example we refer the Reader to [8] or [7].

The distinctive characteristic of Lingo is that it first identifies meaningful cluster labels and only then assigns search results to these labels to build proper clusters. The algorithm consists of five phases. Phase one is preprocessing of the input snippets, which includes tokenization, stemming and stop-word marking. Phase two identifies words and sequences of words frequently appearing in the input snippets. In phase three, a matrix factorization is used to induce cluster labels. In phase four snippets are assigned to each of these labels to form proper clusters. The assignment is based on the Vector Space Model (VSM) [15] and the cosine similarity between vectors representing the label and the snippets. Finally, phase five is postprocessing, which includes cluster merging and pruning.

In the context of this paper, phase three – cluster label induction – requires most attention. This phase relies on the Vector Space Model [15] and a term-document matrix A having t rows, where t is the number of distinct words found in the input snippets, and d columns, where d is the number of input snippets. Each element a_{ij} of A numerically represents the relationship between word i and snippet j. Methods for calculating a_{ij} are commonly referred to as *term weighting schemes*, refer to [15] for an overview. The key component in label induction is an approximate matrix factorisation, which is used to produce a low-dimensional basis for the column space of the term-document matrix.

The motivation behind using the low-dimensional basis for label discovery is the following. In linear algebra, base vectors of a linear space can be linearly combined to create any other vector belonging that space. In many cases, base vectors can have interpretations that are directly related to the semantics of the linear space they span. For example, in [9] an approximate matrix factorisation called Non-negative Matrix Factorisation (NMF) applied to human face images was shown to be able to produce base vectors corresponding to different parts of a human face. It is further argued in [16] that low-dimensional base vectors can discover the latent structures present in the input data. Following this intuition, we believe that in the search results clustering setting each of the base vectors should carry some broader idea (distinct topic) referred to in the input collection

of snippets. Therefore, in Lingo, each vector of the low-dimensional basis gives rise to one cluster label.

Unfortunately, base vectors in their original numerical form are useless as human-readable cluster descriptors. To deal with this problem, we use the fact that base vectors obtained from a matrix factorisation are vectors in the original term space of the term-document matrix. Moreover, frequent word sequences or even single words appearing in the input snippets can also be expressed as vectors in the same vector space. Thus, the well-known measures of similarity between vectors, such as the cosine similarity [15], can be used to determine which frequent word sequence or single word best approximates the dominant verbal meaning of a base vector. Bases produced by particular factorisation methods can have specific properties, discussed below, which can have an impact on the effectiveness of the label induction phase as a whole.

3.2 Matrix Factorisations

To introduce the general concept of matrix factorisation, let us denote a set of d t-dimensional data vectors as columns of a $t \times d$ matrix A.[2] The task of factorisation, or decomposition, of matrix A is to break it into a product of two matrices U and V so that $A \approx UV^T$, the sizes of the U and V matrices being $t \times k$ and $d \times k$, respectively. Columns of the U matrix can be thought of as base vectors of the new low-dimensional linear space, and rows of V as the corresponding coefficients that enable to approximately reconstruct the original data.

Singular Value Decomposition. Singular Value Decomposition (SVD) breaks a $t \times d$ matrix A into three matrices U, Σ and V such that $A = U\Sigma V^T$. U is a $t \times t$ orthogonal matrix whose column vectors are called the left singular vectors of A, V is a $d \times d$ orthogonal matrix whose column vectors are termed the right singular vectors of A, and Σ is a $t \times d$ diagonal matrix having the singular values of A ordered decreasingly. Columns of U form an orthogonal basis for the column space of A. Lingo uses columns of the U matrix to induce cluster labels.

In the context of search results clustering, an important feature of SVD is that the U matrix is orthogonal, which should lead to a high level of diversity among the induced cluster labels. On the other hand, to achieve the orthogonality, some components of the SVD-derived base vectors may have to be negative. This makes such components hard to interpret in terms of their verbal meaning. Moreover, although in practice the cosine distance measure seems to work well in the SVD-based cluster label induction phase, interpretation of the similarity between sequences of words and base vectors would be more straightforward if the latter contained only non-negative values.

Non-negative Matrix Factorisation. The Non-negative Matrix Factorisation (NMF) was introduced in [9] as a means of finding part-based representation of

[2] In the related literature the numbers of rows and columns are usually denoted by m and n, respectively. In this paper, however, we have decided to adopt a convention that directly relates to a term-document matrix having t rows and d columns.

human face images. More formally, given k as the desired size of the basis, NMF decomposes a $t \times d$ non-negative matrix A into two nonnegative matrices U and V such that $A \approx UV^T$, the sizes of U and V being $t \times k$ and $d \times k$, respectively. An important property of NMF is that by imposing the non-negativity constraints it allows only additive, and not subtractive, combinations of base vectors. Lingo will use columns of the U matrix as base vectors for discovering cluster labels.

The non-negativity of the base vectors enables us to interpret the verbal meaning of such vectors in an intuitive way, i.e. the greater value of a component in the vector, the more significant the corresponding term is in explaining its meaning. This also makes the interpretation of the cosine similarity between sequences of words and base vectors less ambiguous. On the other hand, the non-negativity of the NMF-derived basis is achieved at the cost of the base vectors not being orthogonal, which may cause some of the NMF-induced cluster labels to be more similar to each other than desired. In this paper we tested two slightly different variants of NMF described in [16]: NMF with Euclidean distance minimisation (NMF-ED) and NMF with Kullback-Leibler divergence minimisation (NMF-KL).

Local Non-negative Matrix Factorisation. Local Non-negative Matrix Factorisation (LNMF) is a variation of NMF introduced in [10] that imposes three additional constraints on the U and V matrices, which aim to expose the local features of the examples defined in the A matrix. The constraints are: maximum sparsity in V (V should contain as many zero elements as possible), maximum expressiveness of U (retain only those elements of U that carry most information about the original A matrix) and maximum orthogonality of U. Lingo will use columns of the U matrix to discover prospective cluster labels.

Being a variant of NMF, Local Non-negative Matrix Factorisation inherits all its advantages, including the non-negativity of base vectors. Additionally, the fact that LNMF promotes sparseness of the base vectors should result in less ambiguous matching between these vectors and frequent phrases. The special emphasis on the orthogonality of U is also desirable as it guarantees high diversity among candidate cluster labels. A possible disadvantage of LNMF in the context of search results clustering is its slow convergence [10].

Concept Decomposition. Concept Decomposition (CD) [11] is a factorisation method based on the Spherical K-Means clustering algorithm. For a $t \times d$ matrix A and given k, Concept Decomposition generates a $t \times k$ matrix U and a $d \times k$ matrix V such that $A \approx UV^T$. In the CD factorisation, each column of the U matrix directly corresponds to one centroid obtained from the K-Means algorithm. For cluster label induction Lingo will use the U matrix.

Because K-Means is based around averaged centroids of groups of documents, it should be able to successfully detect major themes in the input snippets. However, it may prove less efficient in identifying topics represented by relatively small groups of documents.

There also exists a class of decomposition techniques based on random projections [17]. Even though these decompositions fairly well preserve distances and similarities between vectors, they are of little use in our approach. The reason

is that they rely on randomly generated base vectors, which will directly lead to random labels being induced.

4 Experimental Setup

The primary aim of our experiment was to compare how four different matrix factorisations perform as parts of a description-comes-first search result clustering algorithm. We divided our tests into three parts: topic separation experiment, outlier detection experiment and subjective cluster label quality judgments. The aim of the topic separation experiment was to test the algorithms' ability to identify major topics dealt with in the input snippets. The outlier detection experiment aimed at verifying whether the algorithms can highlight a small topic that is clearly different from the rest of the test set. Finally, we subjectively analysed the properties of cluster labels produced by the algorithms.

4.1 Merge-Then-Cluster Approach Using Open Directory Project

We performed our experiments using data drawn from the Open Directory Project, which is a large human-edited hierarchical directory of the Web. Each branch of the ODP hierarchy, called a category, corresponds to some distinct topic (e.g. "Assembler Programming" or "Stamp Collecting") and contains links to Internet resources dealing with that topic. Every link in ODP is accompanied by a short (25–30 words) description, which in our setting emulates the contextual snippet returned by a search engine.

To implement the merge-then-cluster evaluation, we created 77 data sets, each of which contained a mixture of documents originating from 2 to 8 manually selected ODP categories. In 63 data sets, which were used in the topic separation experiment, each category was represented by an equal number of documents. The remaining 14 data sets, created for the outlier detection experiment, contained equal numbers of documents from 4 closely related ODP categories (major categories) plus documents from 1 or 2 categories dealing with a totally different subject (outlier categories). The numbers of documents representing the outlier categories varied from 100% to 10% of the number of documents representing one major category in that test set. In Table 1 we present an example outlier detection data set containing documents from one outlier category of size 30%. During the experiment, we fed all 77 data sets to the clustering algorithms and compared the contents of the automatically generated clusters with the reference categories defined in ODP.

Reliability of the merge-then-cluster approach largely depends on the way the correspondence between the automatically generated clusters and the original reference groups is measured. The similarity between two sets of clusters can be expressed as a single numerical value using e.g. mutual-information measures [18]. One drawback of such measures is that a smallest difference between the automatically generated clusters and the reference groups will be treated as the algorithm's mistake, even if the algorithm made a different but equally justified choice (e.g. split a large reference group into sub-groups).

Table 1. An example outlier detection test set (four major and one outlier topic)

ODP CatId	Category path	Document count
429194	Computers/Internet/Abuse/Spam/Tracking	28
397702	Computers/Internet/Protocols/SNMP/RFCs	28
791675	Computers/Internet/Searching/.../Google/Web_APIs	28
5347	Computers/Internet/Chat/IRC/Channels/DALnet	29
783404	Science/Chemistry/Elements/Zinc (outlier)	11

To alleviate this problem, we have decided to use alternative measures: Cluster Contamination, Topic Coverage and Snippet Coverage. Due to the limited length of this paper we can only afford an informal description of these measures, we refer the reader to [8] and [19] for formalised definitions.

4.2 Clustering Quality Measures

Let us define the Cluster Contamination (CC) measure to be the number of pairs of documents found in the same cluster K but originating from different reference groups divided by the maximum potential number of such pairs in K. According to this definition, a cluster is *pure* if it contains documents belonging to only one reference group. Noteworthy is the fact that a cluster that consists of only a subset of some reference group is still pure. The contamination measure of pure clusters is 0. If a cluster contains documents from more than one reference group, its contamination measure falls within the 0..1 range. Finally, in the worst case, a cluster consisting of an equally distributed mixture of snippets representing different reference groups will be called *contaminated* and will have the CC measure equal to 1.

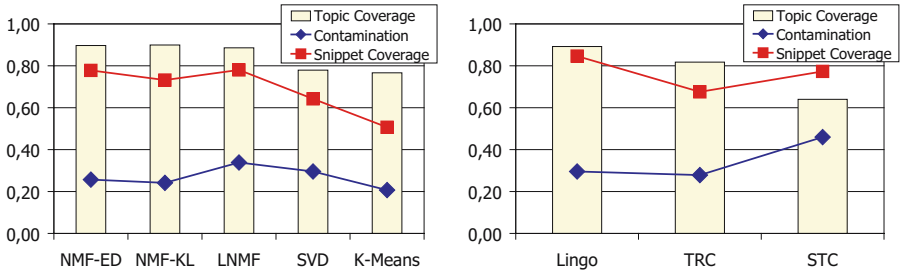
A simple example of a situation where the Cluster Contamination measure alone fails is when for a large number of reference groups the clustering algorithm generates clusters containing documents from only one reference group. In this case Cluster Contamination of all these clusters will be 0, and the algorithm will not get penalized for not detecting topics corresponding the remaining reference groups. To avoid such situations we have decided to introduce a complementary measure called Topic Coverage (TC). TC equal to 1 means that all reference groups have at least one corresponding cluster generated by the algorithm. Topic Coverage equal to 0 means that none of the clusters corresponds to any of the reference groups. Clearly, Topic Coverage promotes algorithms that can create clusters representing both major and outlier topics found in the input set. In our opinion, such behaviour is perfectly reasonable, as it helps the users to find the documents of interest more quickly, even if they come from a small outlier topic.

As clustering algorithms may omit some input snippets or put them in a group of unclustered documents, it is important to define the Snippet Coverage (SC) measure, which is the percentage of snippets that have been assigned to at least one cluster.

5 Experiment Results

5.1 Topic Separation Experiment

Figure 1(a) presents average[3] Topic Coverage, Cluster Contamination and Snippet Coverage for variants of Lingo employing different matrix factorisation algorithms. The NMF-like factorisations provide significantly[4] better average topic and snippet coverage, the difference between the NMF-like algorithms themselves being statistically insignificant. Interesting is the much higher value of cluster contamination in case of the LNMF algorithm compared to the other NMF-like factorisations. We explain this phenomenon when we analyse cluster labels generated by all the algorithms.



(a) Matrix factorisation comparison (b) Clustering algorithm comparison

Fig. 1. Topic coverage, snippet coverage and cluster contamination measures in the topic separation experiment

Figure 1(b) shows how Lingo (NMF-ED) compares with two other search results clustering algorithms that do not follow the description-comes-first paradigm: Suffix Tree Clustering (STC) [13] and Tolerance Rough Set Clustering (TRC) [4]. Compared to TRC and STC Lingo achieves significantly better topic and snippet coverage. TRC produces slightly purer clusters, but the difference is not statistically significant. The above results prove that the description-comes-first approach to search results clustering is a viable alternative to the existing algorithms.

5.2 Outlier Detection Experiment

Table 2(a) summarises the number of outliers detected by variants of Lingo using different matrix factorisations. Interestingly, the base line K-Means-based factorisation did not manage to reveal any of the outliers, neither in the one-outlier data set nor in the two-outlier one. This may be because K-Means tends

[3] For full results, please refer to [19].

[4] Due to the fact that our data does not follow Gaussian distribution, differences marked hereafter as statistically significant have been tested using the Mann-Whitney non-parametric two-group comparison test at the significance level of 0.001.

Table 2. Numbers of detected outliers in the outlier detection experiment. For each matrix factorisation and each clustering algorithm we provide the numbers of detected outliers for data sets containing one and two outliers.

Outlier size	Detected outliers									
	NMF-ED		NMF-KL		LNMF		SVD		K-Means	
	1	2	1	2	1	2	1	2	1	2
100%	1	2	1	2	1	2	1	1	0	0
50%	1	2	1	1	1	1	1	1	0	0
40%	1	2	1	2	1	2	0	0	0	0
30%	1	1	1	1	1	1	0	1	0	0
20%	1	2	1	2	1	2	1	1	0	0
15%	1	1	0	1	0	1	1	2	0	0
10%	1	0	1	0	1	0	1	1	0	0

(a) Matrix factorisation comparison

Outlier size	Detected outliers					
	Lingo		TRC		STC	
	1	2	1	2	1	2
100%	1	2	1	1	1	0
50%	1	2	0	1	0	0
40%	1	2	0	2	0	0
30%	1	1	0	1	0	0
20%	1	2	0	0	0	0
15%	1	1	0	0	0	0
10%	1	0	0	0	0	0

(b) Clustering algorithms comparison

to locate its centroids in most dense areas of the input snippet space, which is usually not where the outliers lie. All NMF-like methods performed equally well, slightly better than SVD. SVD, however, was the only algorithm do discover one of the two smallest 10% outliers.

In Table 2(b) we show how Lingo (NMF-ED) compared with the Suffix Tree Clustering (STC) and Tolerance Rough Set Clustering (TRC) algorithms in the outlier detection task. Clearly, Lingo proves superior to the other two algorithms in this task for both one- and two-outlier data sets. This demonstrates the NMF's ability to discover not only the collection's major topics but also the not-so-well represented themes.

5.3 Subjective Label Quality Judgements

Figure 2 shows the labels of clusters produced by Lingo with different matrix factorisations for a data set containing documents from four ODP topics: Assembler Programming, Oncology, Collecting Stamps and Earthquakes. In the author's opinion, the majority of cluster labels, especially those placed at top positions on the cluster lists, are well-formed readable noun phrases (e.g. "Earthquake Prediction", "Oncology Conference", "Stamp Collecting", "Assembly Language Programming"). One interesting phenomenon is that two very similar labels appeared in the NMF-ED results: "Assembly" and "Assembler Programming". The reason for this is that the English stemmer we used did not recognise the words *assembly* and *assembler* as having the same stem.

A more careful analysis of the cluster labels created by the LNMF version of Lingo can reveal why this algorithm produces significantly more contaminated clusters (compare Figure 1(a)). The key observation here is that LNMF aims to generate highly sparse and localised base vectors, i.e. having as few non-zero elements as possible. This results in a high number of one-word candidate labels, such as "University", "Engineering", "World" or "Network", which in turn contributes to the high cluster contamination.

While cluster labels produced by the K-Means decomposition are generally readable and informative, they only cover the major topics of the test set, which

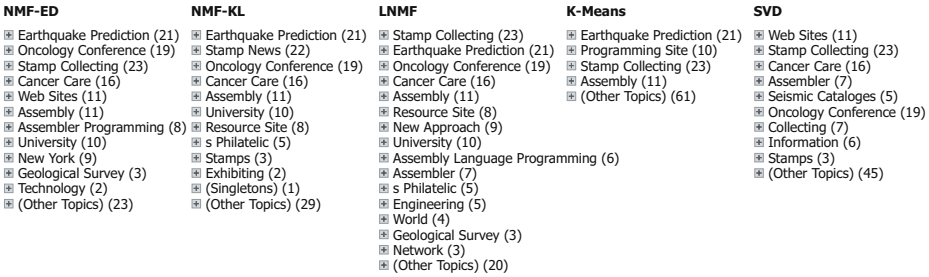


Fig. 2. Matrix factorisation comparison: cluster labels

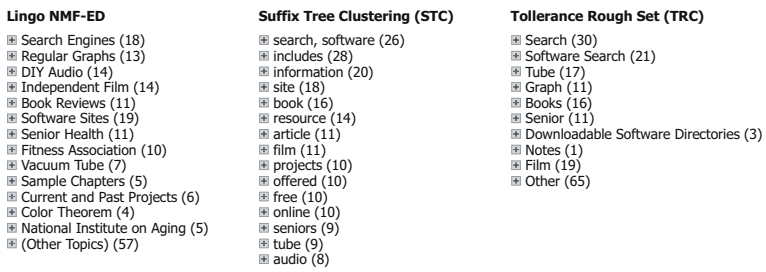


Fig. 3. Clustering algorithm comparison: cluster labels

further confirms poor performance of K-Means decomposition in the outlier detection test.

In Figure 3 we show cluster labels generated by Lingo, STC and TRC for a data set containing six ODP categories: Book Previews, Search Engines, Fitness, Do-It-Yourself, Graph Theory and Independent Filmmaking. Compared to STC and TRC Lingo seems to produce labels that are slightly more specific and probably easier to interpret, compare: "Search Engines" (Lingo) vs. "Search" (TRC), "Vacuum Tube" (Lingo) vs. "Tube" (STC and TRC) or "Independent Film" (Lingo) vs. "Film" (STC and TRC). Also, for this particular data set Lingo managed to avoid generating too general or meaningless labels such as "free", "online", "site", "includes", "information" (STC) or "Notes" (TRC).

6 Conclusions and Further Work

In this paper we have shown how a matrix factorisation can be used as part of a description-comes-first approach to search results clustering. We tested four factorisation algorithms (NMF, LNMF, SVD and K-Means/Concept Decomposition) with respect to topic separation, outlier detection and label quality. We also compared our approach with two other algorithms not based on matrix decompositions: Suffix Tree Clustering and Tolerance Rough Set Clustering.

Our experiments revealed that the Non-negative Matrix Factorisations significantly outperform both SVD and Concept Decomposition with respect

to topic and snippet coverage, while maintaining almost the same level of cluster contamination. The reason for this is that, in contrast to SVD, NMF produces non-negative base vectors which can be better matched with the frequent phrases found in the input snippets. Another important observation is that due to high sparsity of base vectors, Local Non-negative Matrix Factorisation generates cluster labels that are shorter and more general compared to the other NMF methods. For this reason, contrary to our initial expectations, LNMF performed much worse with respect to average cluster contamination, and thus in the present form is not the best choice factorisation algorithm for Lingo. Finally, the description-comes-first approach to search results clustering implemented by Lingo significantly outperformed both STC and TRC in topic separation and outlier detection tests.

We feel that future experiments should investigate more complex matrix factorisations, such as [20]. It is also very interesting how our algorithm would perform for the full-text test collections such as REUTERS-21578 or OHSUMED. Such experiments would require efficient implementations of the factorisations taking advantage of e.g. the high sparsity of the term-document matrix or using subsampling.

Acknowledgment

The author would like to thank anonymous reviewers for helpful suggestions. The experiments were performed within the Carrot^2 Search Results Clustering Framework. Carrot2 is available free of charge from *http://sf.net/projects/carrot2*.

References

1. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1998) 46–54
2. Zamir, O.E.: Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. PhD thesis, University of Washington (1999)
3. Dong, Z.: Towards Web Information Clustering. PhD thesis, Southeast University, Nanjing, China (2002)
4. Lang, N.C.: A tolerance rough set approach to clustering web search results. Master's thesis, Faculty of Mathematics, Informatics and Mechanics, Warsaw University (2004)
5. Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R.: A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proceedings of the 13th international conference on World Wide Web, ACM Press (2004) 658–665
6. Stefanowski, J., Weiss, D.: Carrot^2 and language properties in web search results clustering. In: Proceedings of AWIC-2003, First International Atlantic Web Intelligence Conference. Volume 2663 of Lecture Notes in Computer Science., Madrid, Spain, Springer (2003) 240–249

7. Osiński, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on Singular Value Decomposition. In: Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference. Advances in Soft Computing, Zakopane, Poland, Springer (2004) 359–368
8. Osiński, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems **20**(3) (2005) 48–54
9. Lee, D., Seung, S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
10. Li, S.Z., Hou, X.W., Zhang, H., Cheng, Q.: Learning spatially localized, parts-based representation. In: CVPR (1). (2001) 207–212
11. Dhillon, I., Modha, D.: Concept decompositions for large sparse text data using clustering. Machine Learning **42**(1) (2001) 143–175
12. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In: Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval, Zürich, CH (1996) 76–84
13. Zamir, O., Etzioni, O.: Grouper: a dynamic clustering interface to Web search results. Computer Networks (Amsterdam, Netherlands: 1999) **31**(11–16) (1999) 1361–1374
14. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM Press (2003) 267–273
15. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)
16. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Neural Information Processing Systems. Volume 13. (2000) 556–562
17. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2001) 245–250
18. Dom, B.E.: An information-theoretic external cluster-validity measure. Technical Report IBM Research Report RJ 10219, IBM (2001)
19. Osiński, S.: Dimensionality reduction techniques for search results clustering. Master's thesis, The University of Sheffield (2004)
20. Xu, W., Gong, Y.: Document clustering by concept factorization. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 202–209