

# Using Discrete PCA on Web Pages

W. Buntine, S. Perttu and V. Tuulos

July 26, 2004

# Using Discrete PCA on Web Pages

W. Buntine, S. Perttu and V. Tuulos

Helsinki Institute for Information Technology HIIT

Tammasaarenkatu 3, Helsinki, Finland

PO BOX 9800

FI-02015 TKK, Finland

<http://www.hiit.fi>

HIIT Technical Reports 2004–15

ISSN 1458-9478

URL: <http://cosco.hiit.fi/Articles/hiit-2004-15.pdf>

Copyright © 2004 held by the authors

NB. The HIIT Technical Reports series is intended for rapid dissemination of results produced by the HIIT researchers. Therefore, some of the results may also be later published as scientific articles elsewhere.

Wray Buntine, Sami Perttu and Ville Tuulos

Helsinki Inst. of Information Technology  
P.O. Box 9800, FIN-02015 HUT, Finland  
{firstname}.{lastname}@hiit.fi,  
<http://www.hiit.fi/{firstname}.{lastname}>

**Abstract.** Discrete PCA builds components for discrete data rather like PCA and ICA does for real data. The method has a long history and is most commonly used in genetics. Recent insights into the method are described here, and some examples of given of its use in automatically building a topic model for a document collection, and in its use as a tool for relevance estimation in search. The topic model can also be subsequently used in search. This discussion paper describes our ongoing research here.

**METHOD:** discrete PCA

**PROBLEM:** intelligent search

## 1 Introduction

This paper describes our experiences with the new discrete methods for principle components analysis (PCA) [1, 2], and applies them to the task of relevance in search. These methods are a multinomial analogue to the Gaussian model for probabilistic PCA for instance, see [3]. But they are better described as a discrete version of independent component analysis (ICA) [4]; the connection is proven in [1]. They have been shown to build good probabilistic models of bag-of-word data [5, 6], a model for text, and can be applied more generally to multiple multinomials [7] (for instance, keeping separate multinomials or bags for title words, emphasis words, body text, and link-to text). The most common use of the method is in bioinformatics due to the work of Pritchard *et al.* [7]. Most previous published work on discrete PCA for text, including our own, uses relatively clean text like newspaper articles or scientific abstracts. In this discussion paper we argue that web data needs more sophisticated pre-processing to work well with this method, and show how to use the method as a basis for relevance estimation in search.

## 2 Relevance in Search

There is a strong commercial market and a good base of freeware software for the task of site search or topic specific search. This is the search engine task when restricted either by domain or by topic or sites crawled. Often, these search engines are packaged with a larger corporate intranet suite. One branch of research here is the topic-specific crawler [8]. We are concerned with improving the *relevance* side of this search, in contrast to the *authority* side of search typified by the link-based scores that are the dominant ranking factor for the search

business (such as Pagerank<sup>TM</sup>). Considerable research has extended the popular link-based authority scores to topic-specific authority measures [9], now used in Teoma.

Relevance becomes more important in site search or topic specific search where link information may be poor, hence our interest in it. General articles and expert business opinion on the future of search generally agrees on a typically nebulous statement of the form “understanding the user’s intent.” We note that the user interface, system performance and the selected display of results are just as critical as a quality ranking of results, but we do not consider these issues here, just the relevance of results.

The relevance side of information retrieval is generally considered to be an orthogonal measure to authority. Methods for improving the relevance of retrieved documents have been the subject of the TREC tracks organized by NIST. Until recently the dominant paradigm here was simple versions of TF-IDF, using for instance pseudo-relevance feedback to incorporate empirically related words [10]. A recent promising area is the *language modeling* approach to information retrieval [11], which is based on the simple idea that retrieval should seek to find a document  $D$  that maximizes a probability for the query  $Q$  of  $Pr(Q|D, collection)$ , rather than the earlier notion of  $Pr(D|Q, collection)$  [12]. From a practical viewpoint, this means a change in emphasis from “model the users intent of  $Q$ , and then find matching documents  $D$ ” to “model the content of each document  $D$  and then find queries  $Q$  matching the content”. For the computational statistician, the difference is stark: discrete statistical modelling of documents is feasible whereas modelling of queries of size 2 or 3 is not. Language modeling made its first major applied breakthrough in the 2003 TREC Web track [13], where it ranked a strong first, and additional research [14] suggests the ability to differently model multiple text types (title, heading, body, etc.) is a key factor here.

### 3 Topic Models

Topic categories in news content, for instance “Corporate,” “Government” and “Markets” generally form a partitional (mutually exclusive and exhaustive) space. That is, articles tend to belong to one or a few categories. Correlation co-efficients for categories in Reuters news collections, for instance, are quite negative. This can be explained by the coding methods used at Reuters: an article is assigned its “major category” [15], and sometimes a few others. Partitional category spaces are modelled well by standard text clustering methods which seek to jointly discover a set of categories and assign each document to one of the categories.

Simple empirical tests show that the same partitional tendencies do not hold for web categories such as the Open Directory Project<sup>1</sup> (ODP). Quite a lot of cross linking exists between the nodes of the directory, and queries made to Google’s interface to the ODP typically return 3 or more categories in the top

<sup>1</sup> <http://www.dmoz.org>

10. While one might expect this for categories such as Regional, it also applies to others. With categories such as Kids, Science, Recreation, Computers, Shopping, etc., it is easy to see that categories are somewhat independent. One can mix and match categories quite easily: one can have Kids Science, Recreational Computers, Shopping for Recreational items, Scientific Computing, etc. One exception is Adult which is not fairly represented in the ODP.

Now this independence could be explained in part by the construction process: independent maintainers for the sub-trees of the ODP hierarchy. This independence could also be explained from an information theoretic view-point: independent features form the most efficient access/encoding method. But from our perspective, they also suggest that discrete PCA, which is known to build independent components, is a better topic model than traditional clustering when dealing with web data. Note that discrete PCA methods are easily tweaked to model different text types differently (i.e., separating out title, body, link-to text, etc.) [1], thus we claim they form a better tool for unsupervised topic modelling on the web.

Note the superiority of discrete PCA over clustering from a perplexity view-point (i.e., likelihood scores for the model, where the discovered categories are viewed as irrelevant) has already been shown for text [5].

## 4 The Discrete PCA Topic Model

The discrete PCA topic model is a multi-aspect topic model. It allows multiple topics to co-exist in the one document [5, 6, 2]. The simplest version consists of a linear admixture of different multinomials, and can be thought of as a generative model for sampling words to make up a bag, for the Bag of Words representation for a document [16].

- We have a total count  $L$  of words to sample.
- We partition these words into  $K$  topics, components or aspects:  $c_1, c_2, \dots, c_K$  where  $\sum_{k=1, \dots, K} c_k = L$ . This is done using a hidden proportion vector  $\mathbf{m} = (m_1, m_2, \dots, m_K)$ . The intention is that, for instance, a sporting article may have 50 general vocabulary words, 40 words relevant to Germany, 50 relevant to football, and 30 relevant to people’s opinions. Thus  $L=170$  are in the document and the topic partition is (50,40,50,30).
- In each partition, we then sample words according to the multinomial for the topic, component or aspect. This is the base model for each component. This then yields a bag of word counts for the  $k$ -th partition,  $\mathbf{w}_{k,\cdot} = (w_{k,1}, w_{k,2}, \dots, w_{k,J})$ . Here  $J$  is the dictionary size, the size of the basic multinomials on words. Thus the 50 football words are now sampled into actual dictionary entries, “forward”, “kicked”, “covered” etc.
- The partitions are then combined additively, hence the term admixture, to make a distinction with classical mixture models. This yields the final sample of words  $\mathbf{r} = (r_1, r_2, \dots, r_J)$  by totalling the corresponding counts in each partition,  $r_j = \sum_{k=1, \dots, K} w_{k,j}$ . Thus if an instance of “forward” is sampled

twice, as a football word and a general vocabulary word, then we return the count of 2 and its actual topical assignments are lost, they are hidden data.

This is a full generative probability model for the bag of words in a document. The hidden or latent variables here are  $\mathbf{m}$  and  $\mathbf{w}$  for each document, whereas  $\mathbf{c}$  is derived. The proportions  $\mathbf{m}$  correspond to the components for a document, and the counts  $\mathbf{w}$  are the original word counts broken out into word counts per component.

For the ICA version of this model, the components are represented as  $L\mathbf{m} = (Lm_1, \dots, Lm_K)$ , which are in units of “counts”. Thus we say the document has approximately  $Lm_k$  words counted in component  $k$ , and these estimated counts are independent a-prior.

There are two computationally viable schemes for *learning* these models from data. The mean field approach [6, 17] and Gibbs sampling [7, 2]. Gibbs sampling is usually not considered feasible for large problems, but in this application it can be used to hone the results of faster methods, and also it is moderately fast due to the specifics of the model. Gibbs sampling also seems to create much more accurate component estimates for a document. Extensive experimentation has proven it to be our method of choice.

The *inference task* is, given a particular component model (previously obtained from a full collection via Gibbs or mean-field), estimate the components for a new document. We use the term “estimate” in the statistical sense because the components are a hidden variable, and never known exactly.

## 5 Using Discrete PCA to Organize a Collection

The discrete PCA models by themselves have been seen by us to be a good approach for organizing a document collection. We give two examples here.

### 5.1 Human Rights

We crawled a small collection of 18,000 HTML documents on human rights from high profile human rights organizations (Red Cross, MSF, UN, etc.). We restricted each document to its first 1000 words (for reasons to be discussed in Section 6). Building discrete PCA components with  $K=30$  components took a few hours on a 3GHz CPU, and then one of us named them manually in two hours using a display tool.

This yields the following organizational structure. Here we have placed different components, each name terminated with a “;”, into groups. The group names, in full upper case, and these groups themselves were created by us for explanation.

**ORGANIZATIONS:** Amnesty International; Human Right Watch; ICC Campaign; Médecins Sans Frontières (MSF); Red Cross; Treaty Bodies Database  
**ACTIONS and the PRESS:** Briefings and the Press; Fear for Safety Alert; Medical Letter Writing; Urgent Action

**COUNTRIES:** African Conflict; China and Far East Reports; Middle East and Myanmar Concerns; Turkey (and nbrs.) Ill-treatment; Former Yugoslavia  
**PROGRAMS and AFFAIRS:** World Economy; American vs. European affairs; American and its Programs;  
**VICTIMS:** Attacks on Civilians; Children, and Women; Conscientious Objectors; Death Penalty in the USA; Disappeared and Forgotten, Legacy and Justice; Prisoners; Refugees; Workers Rights, Korea and elsewhere  
**INFRASTRUCTURE:** Elections and Political Process; Justice; Preparedness and Threats; Police and Force

You can see here in the ICA aspect of discrete PCA emerging. One can almost create joint topics with formulas such as

- ORGANIZATION+COUNTRY+VICTIM,
- ACTION+COUNTRY +VICTIM,
- PRESS+COUNTRY +INFRASTRUCTURE.

We argue this is an informative guide to the content of the collection.

## 5.2 The EU and the UN

We crawled a collection of approximately 230,000 HTML and 15,000 PDF documents from 28 EU and UN related sites. Linguistic preprocessing was as follows. The 50 major stop words (including all word classes except nouns) were eliminated. Only the top 3000 numbers were included (tokens such as "1996", "third", "20/20", etc.). Words with less than 5 instances or occurring in less than 3 documents were removed. We have an extended version of discrete PCA that builds hierarchical topic models automatically [1]. The model was built in phases: (1) the top level of 10 nodes and the root, (2) the second level of 10 sets of 10 nodes for the above, (3) and then free floating expansion of subsequent nodes using a branching factor of 5 once the parent node had stabilized. Thus the top two levels are balanced with a branching factor of 10, and the subsequent levels are unbalanced with a branching factor of 5. The final model had 511 components in total. This took 50 hours of time on a dual CPU with 3GHz processors and 2GB of memory. Some detail of the topic hierarchy are given in the Tables 1–4.

In this case manual naming was not done, as in Section 5.1. The phrase summaries for these topics have been entirely automatically generated by looking for distinctive nominal phrases appearing in documents associated with a topic. Full automatic naming of the topics is not feasible: the meaning of a topic is essentially its documents and summarization of documents is a hard task, requiring a deep understanding of the semantics of text. Thus we use phrase summaries instead, which provide good overviews of the topics.

The hierarchy has two top levels with a branching factor of 10, resulting in 111 top nodes, and subsequent nodes have a branching factor of 5. Shown are the top level Nodes 1–10, the children of Node 3, 3.1–3.10, the children of node 3.1, 3.1.1–3.1.5, and the children of node 3.2, 3.2.1–3.2.5. In most cases, the phrase

1	programme; rights; people; States; Conference; world; Nations; Council; women; region;
2	Council; Europe; groups; Commission; European Union; Council of Europe; European Parliament; drugs; European Agency; European Convention;
3	countries; development; people; policies; world; society; population; Office; study; Union;
4	States; members; services; Union; rights; community; Member States; EU; European Union; case;
5	system; activities; project; network; sustainable development; water; European Environment Agency; European Topic Centre; Research Networks;
6	information; products; site; section; documents; list; United Nations; staff; Information Services; web site;
7	Agency; Phone; environment; Denmark Phone; Environment Agency; European Environment Agency; industry; production; report; companies;
8	development; information; programme; project; issues; technology; partners; trade; investment; Institute;
9	years; data; Article; agreement; persons; rate; education; Government; \$; Act;
10	development; States; policies; years; report; meeting; Commission; Committee; action; services;

Table 1. Top Level Topics

summaries are quite clear, though they might be confusing to the general public. Nevertheless, this demonstrates our model building technology, and we believe these would make a strong topic hierarchy for browsing and indexing documents.

## 6 Problems on Web Data

When building and inspecting the models just described, it became clear that web data provides a real challenge to the discrete PCA approach to unsupervised classification. We encountered problems we had not previously encountered with the Reuters collections and so forth. We briefly present them here. We have encountered all these effects on a “Search Engine” document collection built concurrently with the Human Rights collection. Thus we consider the effect to be intrinsic to some web pages. web domain, and in hindsight “obvious.”

First, documents themselves are sometimes are digests, and not on one coherent mixture of topics. For instance, the document may be 10 news articles appended together, unrelated excepting that they occurred in the same day/week. Each segment might be 500 words long making a full document length of 5000 words effectively on many different topics. Strictly speaking, this is a document with mixed topic, thus it should fit the model. In reality, each segment of the document is a completely different mixture of topics. This effect resulted in difficult to interpret components, components we call “junk,” because they do not make any sense. Moreover, the component decompositions  $Lm$  for such documents poorly characterize any of the individual segments of text inside the document. The decompositions are unusable.

3.1	project; Republic; assistance; funds; monuments; contribution; programme; donors; building; disasters;
3.2	schools; students; study; pupils; University; book; primary schools; films; secondary schools; grade;
3.3	cities; Asia; areas; development; town; settlement; authorities; habitats; local authorities; region;
3.4	European Commission; Delegation; Union; European Union; EU; European Commission's Delegation; Europe; relations; co-operation; Member States;
3.5	America; agriculture; countries; Latin America; developing countries; economy; farmers; Caribbean; world; system;
3.6	population; families; Centre; poverty; education; family planning; Philippines; Conference on Population;
3.7	century; links; Africa; media; site; Partner Institutions; Links with Partner Institutions; UNESCO; journalists; Biosphere reserves;
3.8	Delegation; children; people; Head; Chairman; President; elections; room; young people; parties;
3.9	University; Science; Office; Director; team; technology; Professor; Box; UNEP; Library;
3.10	per cent; China; development; goods; period; services; training; administration; economic growth;

**Table 2.** Mid Level Topics under 3 "Countries, Development, People, Policies"

For this reason, we truncated all human rights documents to 1000 words. This dramatically improved the interpretability of the components. We note that many good quality text segmentation algorithms exist, some specialized to web page structure [18, 19]. We need to integrate these kinds of segmentation algorithms as a pre-processing step in our system.

A second problem we encountered is the effect of boilerplate content, for instance the menu bars, site navigation aids, etc., found all around a given site. In some sites, the boilerplate content might be 95% of the actual web-page content, and thus make the analysis poor. Some components will emerge which model the boilerplate for one site as the dominant feature, but it is done poorly and mixes in other artifacts corrupting the full model. Statistically, bag-of-words methods such as discrete PCA are very poor at separating boilerplate from content. Boilerplate recognition algorithms typically make strong use of structure and sequence of tokens [20], all lost in the bag-of-words representation of discrete PCA. Thus we need to preprocess web pages using these methods to remove the annoying artifacts boilerplate creates.

Note there is an interesting contrast here. Our primary role for the discrete PCA model here is in supporting search, and the following sections show. Traditional keyword search with promotion for proximity manages to avoid these two problems quite easily (boilerplate is dealt with by only retaining the best 1-2 matches for a site).

3.1.1	Republic; monuments; Yugoslav Republic; former Yugoslav Republic; phase; People's Republic; Democratic Republic; cultural heritage; Islamic Republic; Republic of Macedonia;
3.1.2	funds; contribution; income; ECHO; total cost; Trust Fund; credit; volunteers; region; Development Fund;
3.1.3	donors; countries; disasters; Iran; cooperation; natural disasters; Democratic Republic; Democratic People's Republic;
3.1.4	building; community; programme; Department; latest major documents; emergency; UNICEF; Emergency Report; WFP Emergency Report;
3.1.5	resources; Coordinator; assessment; contribution; forestry; consortium; technical assistance; preparation; June; Burundi;

**Table 3.** Low Level Topics 3.1 "Projects"

3.2.1	mission; University; programme; activities; Yearbook; High Representative; rehabilitation; programs; crafts; higher education;
3.2.2	supply; images; electricity; water supply; Office; metric tons; cereals; food supplies; urban areas; energy supply;
3.2.3	students; book; project; minorities; training; Association; young people; national minorities; English; members;
3.2.4	TV; audience; TV channels; TV equipment transmissions facilities; audience market share Volume; TV production volume; TV programming; satellite TV channels; TV fiction; Social Council;
3.2.5	study; degree; publication; case studies; grade; population; cities; rural areas; comparative study; Arabic version;

**Table 4.** Low Level Topics 3.2 "Schools"

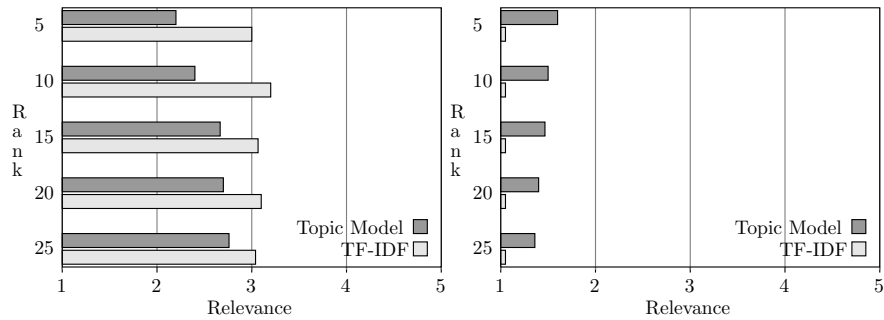
## 7 Using Discrete PCA for Reranking

For relevance testing in the language modeling approach to information retrieval [11], the models are too non-specific. A particular query is a highly specific task and a general statistical model lacks the nuances required to do successful relevance evaluation on that unique query. Thus we instead adopt an approach akin to pseudo-relevance feedback [10]. We take the top 300 documents from the TF-IDF results and build a specific Multinomial PCA model for that subset of documents. We then evaluate the formula  $p(Q|D, \text{sub-collection})$  for the query  $Q$  represented as a bag of words, for each of the top 300 documents  $D$ , and the sub-collection of 300 documents providing the Multinomial PCA components and their word distributions. Techniques for doing this are given in [1]. Thus we build a query specific model and use it. Note, this approach is shown here as a proof of concept; performance issues are not considered.

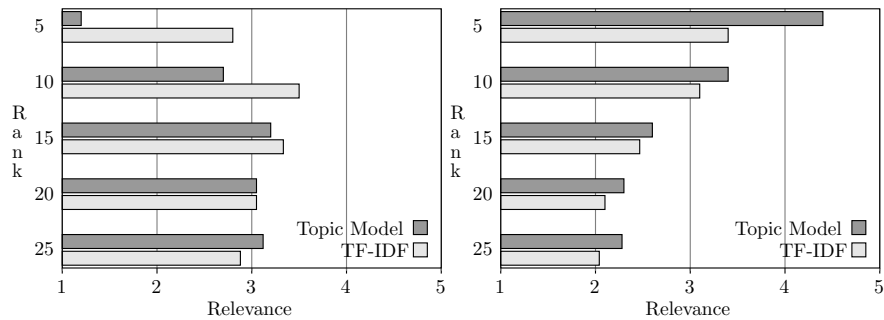
We took EU and UN relevant queries from the TREC *ad hoc* query set. We used queries 401, 404, 407, 409, 410 and 412 as the first 6 queries in the 401-450 set relevant to EU or UN. Queries cover topics such as poaching on wildlife reserves, the Ireland peace problem, and the Schengen agreement. We used the title and description fields as the query words. We ran these queries

through our standard Lemur-like TF-IDF evaluation, and using the Multinomial PCA relevance evaluation. The top ranked 25 results from each were then rated together in a blind test so the rater had no knowledge of the method. Rating used the scale 1-5, with higher being better.

Comparative results are given in Figures 1-3. The bars show the average



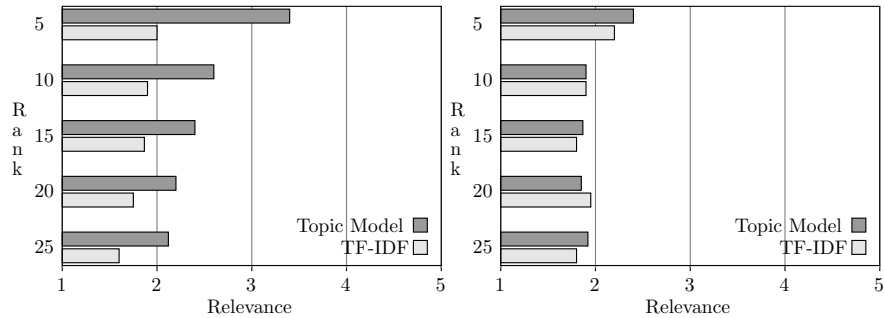
**Fig. 1.** Query 401 and 404 comparisons.



**Fig. 2.** Query 407 and 409 comparisons.

relevance of both methods at ranks 5, 10, 15, 20 and 25, i.e., average relevance at rank 5 is the average relevance of the top 5 documents as rated by the method. The topic model is noticeably better than TF-IDF in 3 queries (404, 409, 410); the methods are about equal in queries 407 and 412; and TF-IDF is better in query 401.

Because the system built a component model on the 300 documents, we can display these components and allow the user to specialize their query to one component or another. This turns out to be very interesting. For the poaching query,



**Fig. 3.** Query 410 and 412 comparison.

only one component in the 10 generated corresponds to relevance to the query, whereas in the Schengen agreement query, several components correspond. The TREC queries are very specific. In more general queries, multiple components indicate the query appears to have multiple responses and should be specialized. We are discussing the available analysis with user-interface experts to attempt to develop an appropriate presentation for harnessing the technique in context.

## 8 Using Discrete PCA for “Context by Example”

Our second application of discrete PCA in search builds on the observation that discrete PCA works well as a language model for information retrieval on rather general queries. Thus, we split our task into the general “context” part, and the specific part.

Many typical information retrieval tasks involve both highly specific elements and a more generic context. Consider, say, query “foreign politics Bush” which should return documents about foreign affairs as pursued by President Bush. A keyword-based search engine wouldn’t probably succeed well in the task since the relevant documents don’t necessarily contain the exact words “foreign politics”. On the other hand, a bare PCA based ranking might return documents on the foreign affairs or politics in general which might not be about President Bush at all. In this case “Bush” represents the specific element of the query and “foreign politics” the context where it should appear.

We solve the issue by making the distinction clear between the specific keywords and their context. Our *Soopa* interface contains a text field for the keywords, as for instance in Google, but in addition there’s a larger box where user can optionally specify a context for the keywords. Naturally the context may be more vague than the keywords since we don’t try to find matches for the context words *per se*. Thus it’s possible to use even an example document as the context specification. Only documents returned by the first keyword search are ranked for context as well.

In practice, *Soopa* works as follows. One tries regular keyword search. If this fails, then some context text is entered into the second box, and this is used to focus the context of the results for the keyword query. The context text is processed by doing a match to a previously built discrete PCA model, for instance those discussed in Section 5. Currently if multiple keywords are provided, we find the intersection of the matching documents i.e., boolean AND. This ensures that the results represent well the specific part of the query. We estimate topic distribution with discrete PCA for the context and use it to rank the corpus subset matching the keywords. Thus the user is shown all the documents which match her keywords, ordered by her context of interest.

Our preliminary tests show that *Soopa* succeeds well in information retrieval tasks where both the content-based ranking and keyword matching alone have difficulties. It provides an easy way for a user to specify context without knowing anything at all about the specific topic hierarchy that is being used to support the context search. The approach, for instance, would work equally well using a topic model built in a supervised fashion after the ODP hierarchy.

Note that the *Soopa* approach is somewhat complementary to the reranking scheme presented in the previous section. The reranking approach builds a new model for a matching corpus subset, making it suitable, say, for result visualization. This is naturally computationally quite demanding and thus might not be suitable for performance sensitive search applications, whereas *Soopa* scales well since the context-based ranking may be implemented efficiently.

## 9 Conclusion

Previous analysis and empirical results have shown that discrete PCA performs well in statistical modelling of bag-of-words data. In this paper we teased out some aspects of this model as it might apply to the search task of retrieving more relevant documents:

- Unsupervised topic hierarchies can be developed, though web data has two significant hurdles, digest pages and boilerplate content, and these need to be preprocessed using known techniques.
- Note we have not discussed the topic/component naming problem here, a subject worthy of another paper.
- Experiments here suggest discrete PCA models might be useful in a post-processing phase for reranking data retrieved using conventional methods. More extensive testing is needed here.
- Discrete PCA models work well in identifying general context, but not in highly specific queries. Thus we can use them in assisting the user focus their queries with a separate “context by example” interface.

**Acknowledgements.** This work was supported in part by the National Technology Agency, under the project Search-Ina-Box, and by the Academy of Finland, under the project Prose. This work was also supported in part by the IST

Programme of the European Community, under the Alvis project and the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

## References

- [1] Buntine, W., Jakulin, A.: Applying discrete pca in data analysis. In: UAI-2004, Banff (2004)
- [2] Griffiths, T., Steyvers, M.: Finding scientific topics. PNAS Colloquium (2004)
- [3] Tipping, M., Bishop, C.: Mixtures of probabilistic principal component analysers. *Neural Computation* **11** (1999) 443–482
- [4] Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley & Sons (2001)
- [5] Hofmann, T.: Probabilistic latent semantic indexing. In: *Research and Development in Information Retrieval*. (1999) 50–57
- [6] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *Journal of Machine Learning Research* **3** (2003) 993–1022
- [7] Pritchard, J., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. *Genetics* **155** (2000) 945–959
- [8] Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery. In: *8th World Wide Web*, Toronto (1999)
- [9] Haveliwala, T.: Topic-specific pagerank. In: *11th World Wide Web*. (2002)
- [10] Singhal, A., Kaszkiel, M.: A case study in web search using TREC algorithms. In: *Proc. of WWW10*. (2001)
- [11] Ponte, J., Croft, W.: A language modeling approach to information retrieval. In: *Research and Development in Information Retrieval*. (1998) 275–281
- [12] Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments - part 2. *Information Processing and Management* **36** (2000) 809–840
- [13] Craswell, N., Hawking, D.: Overview of the TREC 2003 web track. In: *Proc. TREC 2003*. (2003)
- [14] Nallapati, R.: Discriminative models for information retrieval. In: *ACM SIGIR Conference*. (2004)
- [15] Lewis, D., Yand, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* **5** (2004) 361–397
- [16] Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison Wesley (1999)
- [17] Buntine, W.: Variational extensions to EM and multinomial PCA. In: *ECML 2002*. (2002)
- [18] Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. *Machine Learning* **34** (1999) 177–210
- [19] Chakrabarti, S.: Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In: *10th World Wide Web*. (2001)
- [20] Bar-Yossef, Z., Rajagopalan, S.: Template detection via data mining and its applications. In: *11th World Wide Web*. (2002)
- [21] Zhai, C.: Notes on the Lemur TFIDF model. Note with lemur 1.9 documentation, School of CS, CMU (2001)