Computing and Informatics, Vol. 33, 2014, 652–666

COMPARISON OF LATENT SEMANTIC ANALYSIS AND PROBABILISTIC LATENT SEMANTIC ANALYSIS FOR DOCUMENTS CLUSTERING

Marcin Kuta, Jacek Kitowski

AGH University of Science and Technology
Department of Computer Science,
Al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: {mkuta, kito}@agh.edu.pl

Abstract. In this paper we compare usefulness of statistical techniques of dimensionality reduction for improving clustering of documents in Polish. We start with partitional and agglomerative algorithms applied to Vector Space Model. Then we investigate two transformations: Latent Semantic Analysis and Probabilistic Latent Semantic Analysis. The obtained results showed advantage of Latent Semantic Analysis technique over probabilistic model. We also analyse time and memory consumption aspects of these transformations and present runtime details for IBM BladeCenter HS21 machine.

Keywords: Document clustering, latent semantic analysis, probabilistic latent semantic analysis, natural language processing

Mathematics Subject Classification 2010: 68T50, 68T05, 68T35

1 INTRODUCTION

Document clustering refers to all techniques of dividing documents into groups having similar characteristics called clusters. Obtaining such groups is an important issue in information retrieval, topic extraction, and web search results organizing and filtering [6, 18]. Document clustering may also be helpful as a part of plagiarism detection systems. Our experience with processing Polish shows that traditional clustering algorithms like k-means or agglomerative algorithms achieve relatively low accuracy when applied to documents clustering [15]. The reason of this drawback is that these algorithms operate on documents represented by points in highly dimensional space. In the effect computing distance between points becomes problematic and each pair of documents shows very little similarity. This results in clusters of bad quality. One way of improving clustering results is application of dimensionality reduction techniques [15].

This article compares the characteristics of two variants of dimensionality reduction techniques: Latent Semantic Analysis (LSA) [4] and Probabilistic Latent Semantic Analysis (PLSA), which were applied to improve results of partitional and agglomerative clustering algorithms. Accuracy of these algorithms was verified on a corpus of Polish news articles. Parameters taken into account during tuning above algorithms to Polish are also presented. To our knowledge no such comparison was done for Polish documents clustering before.

LSA exploits Singular Value Decomposition (SVD) of document-term matrices. However, this transformation has two drawbacks:

- it does not define properly normalized probability distribution,
- matrices obtained during SVD may contain negative entries.

PLSA [9] alleviates these problems, as it is based on firm statistical foundations – it defines correct probability distribution over documents and terms. PLSA provides better interpretation of dimensions in latent spaces than LSA.

Achieved accuracy is not the only factor, which determines the choice of the right dimensionality reduction method. Building a language model is characterized by time consuming computations and large memory requirements. These demands still grow when it comes to processing inflective languages, like Polish [16]. We compared time and memory consumption of both dimensionality reduction approaches.

2 STATE OF THE ART

The previous works on the subject of document clustering focused on the English language. A wide range of algorithms has been developed: partitional methods, agglomerative methods, model based, density based and others.

Partitional clustering algorithms belong to a family of non-hierarchical algorithms. They start with a set of initial clusters selected in a random way, which are then refined iteratively according to a clustering criterion function. Partitional algorithms include k-means and its variants like bisecting k-means or vector space k-means.

Agglomerative algorithms [22] are an example of hierarchical algorithms. Within this approach initially each point forms a separate cluster. They are iteratively merged until the desired number of clusters is reached. ROCK (RObust Clustering using linKs) [7] is an instance of hierarchical agglomerative algorithm, which uses links instead of distance metrics when merging clusters. Agglomerative clustering generally produces worse results than partitional algorithms [22].

Papers [21, 22] examine agglomerative and partitional algorithms applied to wide range of English texts: TREC test collection (including extracts from LA Times and San Jose Mercury), Reuters-21578 test collection and OHSUMED-23345 dataset.

Self Organising Maps (SOMs), representatives of model based clustering, are a form of neural networks used for feature reduction. SOMs map documents onto two-dimensional document map. GHSOM (Growing Hierarchical SOM) extends the SOM approach and solves the problem of the a priori definition of the map structure [12].

Density based clustering methods are based on a concept of density, defined as a number of points which lie in a certain neighbourhood of considered point. Clusters are formed from areas of high density separated by regions of sparser density. The main example of density based approach is DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm. However, despite its ability to find non convex clusters it did not gain much popularity in documents clustering [5].

Feature and dimensionality reduction techniques which can be applied to documents clustering include Principal Component Analysis (PCA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation, and concept indexing (CI) [11].

The works done in the context of Polish are rare. They covered GHSOM and ROCK algorithms [2], which were used for clustering documents from the IPI-PAN corpus and *Dziennik* newspaper and then for thesaurus extraction. The other method examined was LSA [15, 17].

3 DATASET

The corpus of articles from Polish newspaper *Rzeczpospolita* [19] was used as a dataset in our clusterization experiments. The corpus was subjected to a series of preprocessing transformations, including stopwords filtering, part-of-speech (POS) tagging with the pretrained Trigrams'n'Tags (TnT) tagger [1, 14] emitting IPIPAN tagset tags. POS tags helped to distinguish between occurrences of nouns, verbs, adjectives, etc. Providing information about case, number, gender and other morphological categories POS tags helped also in transformation of each occurrence of inflected word form to its base form (lemmatisation). Lemmatisation phase was done with morphological analyser of Polish, *Morfeusz* [20]. An alternative approach to inflected forms reduction, stemming, can be found in [13].

We extracted three subsets of articles varying in size and containing 1 000, 10 000 and 20 000 articles respectively. Extracted articles belonged to six disjoint categories (culture, economy, law, national news, international news, and sport) which served as a clustering criterion.

Four representations of each subset of articles were created, namely base, noun, bigram and trigram representation. The base representation includes all words present in a document (transformed to a base form) along with their frequency in a document. The noun representation is restricted to occurrences of nouns. The bigram representation enriches the base representation with occurrences of bigrams. In a similar way, the trigram representation extends the bigram representation with occurrences of trigrams. Only bigrams and trigrams which occurred at least four times in a dataset are taken into account. As an additional condition, only bigrams and trigrams composed from nouns, adjectives, verbs or adverbs (possibly intermixed) are represented.

The total number of terms in a dataset depends on the number of included documents and its representation, which is illustrated in Table 1.

	Number of documents		
	1 000	10 000	20 000
base	34 492	136 058	209 284
noun	19 907	81 774	128 778
bigram	57 874	181 659	298 610
trigram	61 974	192 823	322 593

Table 1. Number of terms as a function of corpus size and corpus representation

4 CLUSTERING AND DIMENSIONALITY REDUCTION

Let us assume the following notation:

- N – the number of documents in a clustered dataset,
- M – the number of different terms present in a dataset,
- K – the number of clusters, equal to the number of latent variables.

At first, each dataset was represented, according to the Vector Space Model (VSM), in a highly dimensional space by a term-document matrix $W = [w_{ij}]_{M\times N}$, where w_{ij} corresponds to the occurrences of i^{th} term in j^{th} document. The number of dimensions of the VSM space corresponds to M – the number of different terms in a dataset. Elements of term-document matrix W were fixed according to one of two weighting schemes: tfidf and logent:

$$\text{tfidf}(d,w) = n(d,w)\cdot\log\frac{N}{df(w)}, \tag{1}$$

$$\text{logent}(d,w) = n(d,w)\cdot\left(1+\sum_{i=1}^{N}\frac{\frac{n(d_i,w)}{n(w)}\log\frac{n(d_i,w)}{n(w)}}{\log N}\right), \tag{2}$$

where

- N – number of documents,
- $n(d,w)$ – number of occurrences of term w in document d,
- $n(w)$ – total number of occurrences of term w in all documents,

- $df(w)$ – number of documents containing term w.

These schemes were used with the VSM model and further with LSA. For PLSA simple word counting was applied, which is explained further in the text. As a similarity measure, *sim*, cosine distance was used.

We examined four clustering algorithms: three partitional algorithms (k-means, bisecting k-means, bisecting k-means with refinements) and the agglomerative algorithm [22]. Clusterization was driven by a criterion function, of which we considered the following: $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{E}_1$, $\mathcal{G}_1$, $\mathcal{G}'_1$, $\mathcal{H}_1$, $\mathcal{H}_2$, single link (*slink*), complete link (*clink*) and unweighted pairwise group method with averages (*upgma*) [21]. The most promising functions, $\mathcal{I}_1$ and $\mathcal{H}_1$, are defined as follows ($\mathcal{H}_1$ makes use of $\mathcal{E}_1$ function):

$$\begin{aligned}
\mathcal{I}_1 &= \max \sum_{i=1}^{k} \frac{1}{|S_i|} \left(\sum_{d,d' \in S_i} sim(d,d') \right), & (3)\\
\mathcal{E}_1 &= \min \sum_{i=1}^{k} |S_i| \frac{\sum_{d \in S_i, d' \in S} sim(d,d')}{\sqrt{\sum_{d,d' \in S_i} sim(d,d')}}, & (4)\\
\mathcal{H}_1 &= \max \frac{\mathcal{I}_1}{\mathcal{E}_1}, & (5)
\end{aligned}$$

where

- $sim(d,d')$ denotes similarity between documents d and d',
- k denotes the current number of clusters,
- S_i – i^{th} cluster of cardinality $|S_i|$ and
- S – the set of all documents.

The other functions we examined were:

$$\begin{aligned}
\mathcal{I}_2 &= \max \sum_{i=1}^{k} \sqrt{\sum_{u,v \in S_i} sim(u,v)}, & (6)\\
\mathcal{G}_1 &= \min \sum_{i=1}^{k} \frac{\sum_{u \in S_i, v \in S} sim(u,v)}{\sum_{u,v \in S_i} sim(u,v)}, & (7)\\
\mathcal{G}'_1 &= \min \sum_{i=1}^{k} n_i^2 \frac{\sum_{u \in S_i, v \in S} sim(u,v)}{\sum_{u,v \in S_i} sim(u,v)}, & (8)\\
\mathcal{H}_2 &= \max \frac{\mathcal{I}_2}{\mathcal{E}_1}, & (9)
\end{aligned}$$

Investigated criterion functions specific to agglomerative algorithms were the following:

$$\begin{aligned} \mathit{slink} &= \max_{u \in S_i, v \in S_j} \mathit{sim}(u,v), & (10) \\ \mathit{clink} &= \min_{u \in S_i, v \in S_j} \mathit{sim}(u,v), & (11) \\ \mathit{upgma} &= \frac{1}{|S_i| \cdot |S_j|} \sum_{u \in S_i, v \in S_j} \mathit{sim}(u,v). & (12) \\ & & (13) \end{aligned}$$

As Table 1 shows, dimensionality of considered datasets is quite high, with over 300 000 dimensions for the biggest dataset represented with trigram model. Results obtained with the above algorithms inclined us to apply two techniques of dimensionality reduction for improving document clustering: Latent Semantic Analysis and Probabilistic Latent Semantic Analysis. These models assume that the text contains hidden semantic structure, disturbed by synonymy and polysemy phenomena.

4.1 Latent Semantic Analysis

LSA is an algebraic approach to dimensionality reduction. LSA projects documents vectors located in a terms space onto a low dimensional space composed from semantic dimensions. The number of semantic dimensions is much lower than the number of terms. While documents (vectors) are sparsely distributed in the original space and most of their coordinates are zero elements, they are much densely spaced in the transformed space. LSA chooses projection with maximal variance of data along axes. Two documents may be very similar in a new space, even though originally they contain few or no common terms.

LSA transformation begins with Singular Value Decomposition (SVD) of a term-document matrix W of size $M \times N$ into three matrices U, S, V^T:

$$W = USV^T, \tag{14}$$

where R is the rank of matrix W, S denotes $R \times R$ diagonal matrix containing R singular values $s_1 \geq s_2 \geq \cdots \geq s_R > 0$ on its diagonal. Singular values of matrix W are defined as square roots of eigenvalues of matrix W^*W (W^* denotes an adjoint matrix of W). Left singular $M \times R$ matrix U and right singular $N \times R$ matrix V are column-orthonormal and satisfy the condition $U^TU = V^TV = I_R$, where I_R denotes the identity matrix of order R.

Dimensionality reduction from M-dimensional space of terms to k-dimensional space of concepts ($k \ll M$) is achieved by taking only k out of R dimensions (noise reduction), i.e. removing columns $k+1, \ldots, R$ from U, columns and rows $k+1, \ldots, R$ from S, and rows $k+1, \ldots, R$ from V^T. This results in matrices W_k, U_k, S_k, V_k^T.

Matrix W_k is a low-rank approximation of W and can be found by the formula

$$W \approx W_k = U_k S_k V_k^T. \tag{15}$$

4.2 Probabilistic Latent Semantic Analysis

The model of PLSA is an aspect model of latent variables, whose underlying mechanism differs considerably from purely algebraic nature of LSA. The schema of probabilistic latent semantic analysis works as follows:

- choose document d_i with probability $P(d_i)$,
- select latent class z_k with probability $P(z_k|d_i)$,
- generate word w_j with probability $P(w_j|z_k)$.

The PLSA exploits the Expectation-Maximisation (EM) algorithm [3]. Conditional probabilities are determined iteratively as fix points during Expectation (E) and Maximisation (M) steps. Expectation and Maximisation steps are executed until convergence of computed values is achieved or required number of iterations is done.

Expectation step (E-step) computes posterior probabilites of unobserved, latent variables z_k:

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)}, \tag{16}$$

Maximisation step (M-step) updates the probabilities $P(w_j|z_k)$ and $P(z_k|d_i)$:

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i, w_m)P(z_k|d_i, w_m)}, \tag{17}$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)P(z_k|d_i, w_j)}{n(d_i)}. \tag{18}$$

Symbols used in the above formulas (16)–(18) are as follows:

- $n(d_i)$ is the document length, i.e. the number of tokens in document d_i,
- $n(d_i, w_j)$ stands for the number of occurrences of word w_j in document d_i,
- $P(z_k|d_i)$ is the probability d_i belongs to cluster k represented by variable z_k,
- $P(w_j|z_k)$ defines the distribution of words for particular cluster z_k,
- $P(z_k|d_i, w_j)$ defines distribution of clusters for particular document d_i and word w_j.

Finally cluster k is assigned to document d_i according to the magnitude of $P(z_k|d_i)$:

$$k = \arg\max_k P(z_k|d_i). \tag{19}$$

The formulation of EM steps (Equations (16)–(18)) was chosen after paper [9], rather than the alternative formulation provided in [8], as it allows direct and convenient computation of clusters assigned to documents on the basis of $P(z_k|d_i)$ (Equation (19)).

5 CLUSTERING EXPERIMENTS AND RESULTS

5.1 Experiment Setup

Within the experiment we examined three approaches to documents clustering:

1. no dimensionality reduction (agglomerative and partitional clustering operating on Vector Space Model),
2. LSA model (VSM dimensionality reduced with LSA, then agglomerative and partitional clustering algorithms operating on resulting space),
3. PLSA model (separate clustering algorithms are not needed as clusters arise with latent variables).

Experiments with basic partitional and agglomerative clusterization algorithms were run on a sample of 10 000 documents. This dataset size results from limitations of the CLUTO package, which we used for clusterization [10]. The same restriction pertains to experiments with LSA, which performs dimensionality reduction, but then relies on a clustering algorithm to find the clusters. For the PLSA algorithm it was possible to do experiments on larger datasets, containing more than 10 000 documents, as it uses Equation (19) to assign documents to clusters and thus does not need external clusterization algorithms. PLSA model was examined on three sets of documents containing 1 000, 10 000 and 20 000 documents, respectively.

Weighting schemes were applied both for basic clusterization algorithms running in VSM space and for LSA model, as giving better results than simple frequency counts. On the contrary, simple frequency counts were used for PLSA (cf. Equations (16)–(18)) which performed better than with elaborated weighting.

Each approach was verified on all corpus representations described in Section 3, i.e. base, noun, bigram and trigram representations. Each algorithm was run exactly once, separately for each dataset version.

All outcomes found by investigated algorithms are reported in terms of purity:

$$Purity = \frac{1}{N} \sum_{k} \max_{j} |c_k \cap l_j| \ , \tag{20}$$

with N denoting the number of documents, $c_1, c_2, \ldots$ – clusters found by a considered algorithm, and $l_1, l_2, \ldots$ – reference clusters.

5.2 Results

Table 2 presents, for each corpus representation, the best clustering result achieved with LSA model. It also shows corresponding clustering results when no LSA was applied, but with all remaining parameters unchanged (i.e., clustering method, clustering criterion function, weighting scheme). For each corpus representation, the bisecting k-means with refinements turned out to be the method achieving the best results in conjunction with LSA. The rest of considered clustering algorithms (k-means, bisecting k-means, agglomerative) achieved worse results.

Representation	Weigthing	Criterion	Number of	with	without
of corpus	scheme	function	dimensions	LSA	LSA
base	logent	$\mathcal{I}_1$	100	82.71	51.05
noun	tfidf	$\mathcal{I}_1$	100	82.39	52.92
bigram	tfidf	$\mathcal{H}_1$	50	81.43	62.51
trigram	tfidf	$\mathcal{H}_1$	50	81.40	63.17

Table 2. Purity of clustering algorithms obtained with and without LSA. Algorithms were run on a sample of 10 000 documents

Table 3 presents clustering results achieved with the PLSA algorithm. The results are compared according to the number of iterations performed by the EM algorithm.

Representation	Number of EM iterations		
of corpus	50	150	300
	1 000 documents		
base	55.00	62.20	64.80
noun	56.60	58.00	58.40
bigram	48.80	50.60	53.00
trigram	47.70	49.00	51.00
	10 000 documents		
base	65.27	69.58	70.75
noun	60.94	59.92	58.08
bigram	67.22	68.25	68.49
trigram	67.55	67.52	67.63
	20 000 documents		
base	68.29	71.60	72.50
noun	66.91	67.68	67.78
bigram	63.66	66.91	72.68
trigram	64.13	68.89	69.50

Table 3. Purity of PLSA run on three datasets [%]

LSA accounts for a significant increase in clustering purity compared to standard VSM model. PLSA algorithm also does better than clustering algorithms not using any dimensionality reduction, although improvement is not so distinct. As a general rule, purity grows with the increasing number of EM iterations. We found that the number of EM iterations should be 150–300, quantity larger than the 20–50 iterations suggested by Hofmann in [9].

The best results obtained with LSA outperformed those obtained with the probabilistic algorithm. It should be taken into account that the best results shown in Table 2 are the effect of parameters tuning (selection of the weighting scheme, choosing the best clustering method and criterion function, selection of the optimal number of dimensions).

5.3 Time and Memory Performance

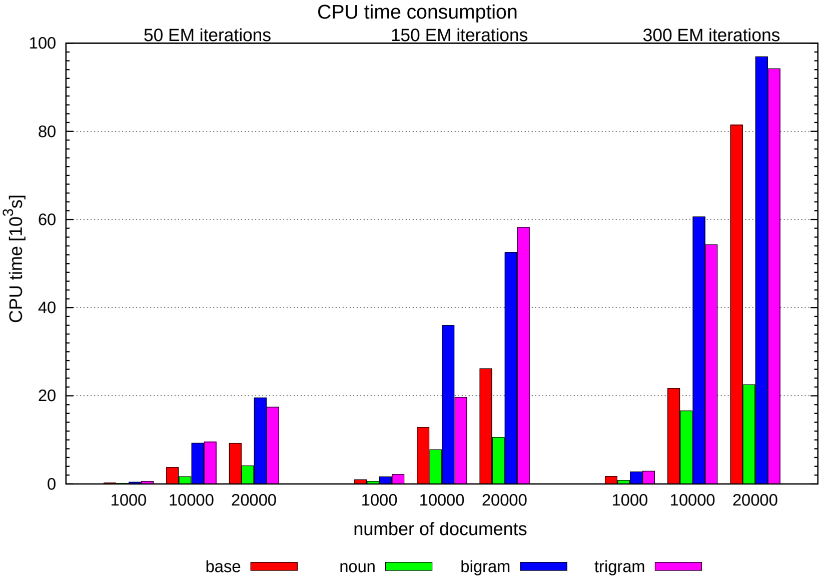


Figure 1. CPU time consumption of the PLSA algorithm vs. the number and representation of clustered documents and the number of EM iterations

Figures 1–4 show time consumption and memory footprint of LSA and PLSA algorithms in our experiments.

Figures 1 and 2 provide comparison of running times of PLSA and LSA algorithms. CPU time of PLSA depends on corpus size, representation of documents (cf. Table 1) and the number of EM iterations. CPU time of LSA clustering (Figure 2) did not depend strongly on corpus representation (e.g. whether it is base, noun, or bigram representation), thus we provided only mean CPU time and its

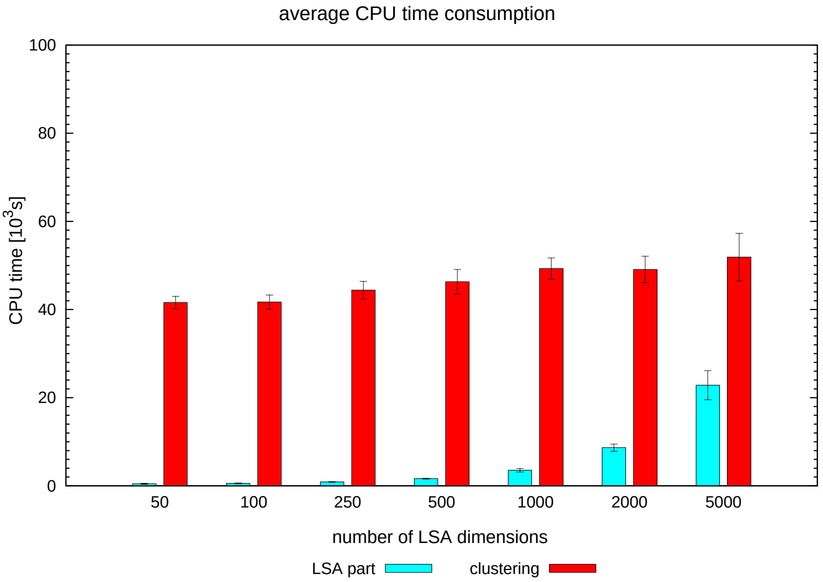


Figure 2. CPU time consumption of the LSA algorithm vs. the number of LSA dimensions

standard deviation over all corpus representations. Execution time of LSA clustering can be decomposed into two elements: time of documents transformation to LSA space (growing with the size of dataset) and relatively constant time of proper clustering.

Figures 3 and 4 provide memory consumption comparison. As can be expected, PLSA memory usage does not depend on the number of EM iterations, but depends overlinearly on the number of documents in a dataset. The reason for this is that the number of unique words in a dataset increases with the growing number of documents. Memory footprint of LSA algorithm did not vary significantly when run on four considered corpus representations. Similarly to Figure 2 we showed mean memory usage along with standard deviation over all corpus representations. It can be inferred from Figure 4 that memory consumption of the SVD algorithm depends heavily on the number of dimensions while memory usage of the proper clustering algorithms does not change with varying number of dimension of LSA space.

Experiments were conducted on an IBM BladeCenter HS21 cluster offering 112 processors of x86-64 architecture. The cluster gave an opportunity to exploit job level parallelism for parameters tuning.

6 CONCLUSIONS

The aim of our paper has been to fill in the gap between research done for English and Polish and provide comparison of latent semantic analysis techniques for Polish.

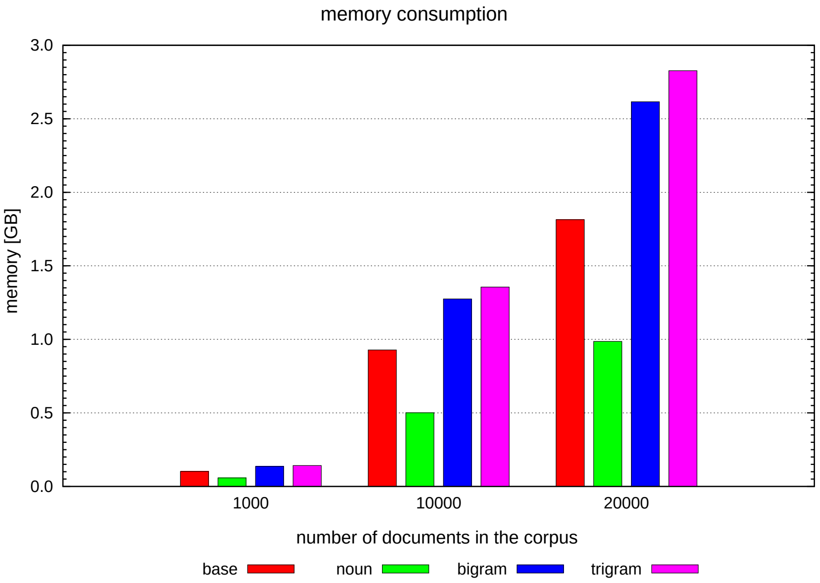


Figure 3. Memory consumption of the PLSA algorithm vs. the number and representation of clustered documents

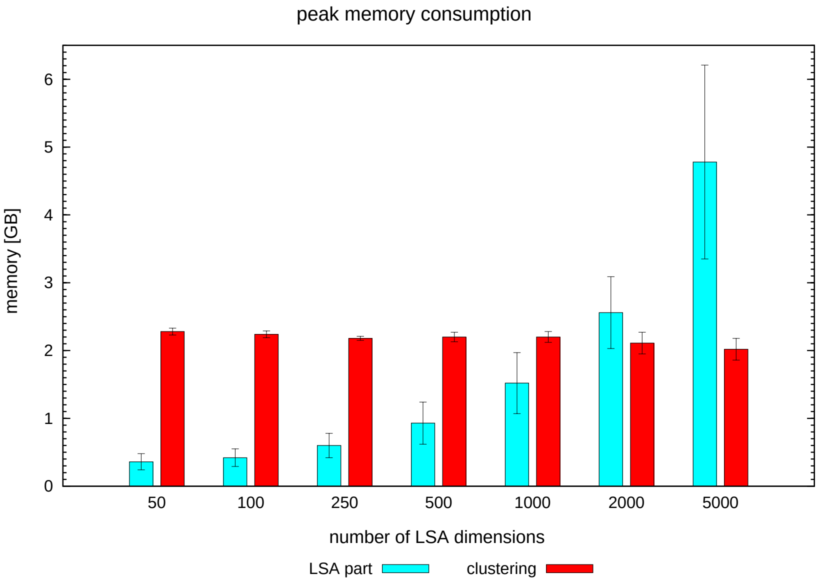


Figure 4. Memory consumption of the LSA algorithm vs. the number of LSA dimensions

Corpus preparation requires taking into account idiosyncrasies of processed language but statistical models of latent semantic analysis themselves are not restricted to any specific language and are applicable to wide range of languages.

Regular partitional and agglomerative clustering algorithms do not provide satisfactory purity of found clusters. Both LSA and PLSA techniques provide significant increase in clusters purity. Contrary to general convictions, LSA performed better than its probabilistic equivalent. Time and memory consumption of LSA depends on different set of parameters in comparison to PLSA, but generally PLSA shows a bit smaller time and memory requirements.

Acknowledgments

This research is supported by UE EFS grant No. UDA-POKL.04.01.01-00-367/08-00. ACC CYFRONET AGH is acknowledged for the computing time.

REFERENCES

[1] Brants, T.: TnT – A Statistical Part-of-Speech Tagger. In: Nirenburg, S., Appelt, D., Ciravegna, F., Dale, R. (Eds.): Proc. of the 6th Applied Natural Language Processing Conference, Seattle, USA 2000, pp. 224–231, 2000.

[2] Broda, B.—Piasecki, M.: Experiments in Documents Clustering for the Automatic Acquisition of Lexical Semantic Networks for Polish. In Proc. of the 8th Int. Intelligent Information Systems Conf., Zakopane, Poland 2008, pp. 203–212.

[3] Dempster, A. P.—Laird, N. M.—Rubin, D. B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, B, Vol. 39. 1977, pp. 1–38.

[4] Dumais, S.: Latent Semantic Analysis. Annual Review of Information Science and Technology, Vol. 38, 2004, No. 1, pp. 188–230.

[5] Ester, M.—Kriegel, H.-P.—Sander, J.—Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Simoudis, E., Han, J., Fayyad, U. (Eds.): Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), Portland, USA, AAAI Press 1996, pp. 226–231.

[6] Gajecki, M.—Krezolek, M.: Automatic Classification of Nouns Into Semantic Groups Using a Corpus of Text. Computer Science, Vol. 5, 2003, pp. 27–39.

[7] Guha, S.—Rastogi, R.—Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes, Information Systems, Vol. 25, 2000, No. 5, pp. 345–366.

[8] Hofmann, T.: Probabilistic Latent Semantic Indexing. In Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, USA 1999, pp. 50–57.

[9] Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, Vol. 42, 2001, pp. 177–196.

[10] Karypis, G.: CLUTO. A Clustering Toolkit. Technical Report 02-017, University of Minnesota, Department of Computer Science 2003.

[11] Karypis, G.—Han, E. H.: Concept Indexing: A Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval & Categorization. Technical Report 00-016, University of Minnesota, Department of Computer Science, Minneapolis 2000.

[12] Kohonen, T.: Self-Organizing Maps (3rd edition). Springer-Verlag, Berlin, ISBN 978-3540679219, 2001.

[13] Korzycki, M.: A Dictionary Based Stemming Mechanism for Polish. In: Sharp, B., Zock, M. (Eds.), Proc. of 9th International Workshop on Natural Language Processing and Cognitive Science, Wrocław, Poland 2012, pp. 143–150.

[14] Kuta, M.—Chrzaszcz, P.—Kitowski, J.: A Case Study of Algorithms for Morphosyntactic Tagging of Polish Language. Computing and Informatics, Vol. 26, 2007, No. 6, pp. 627–647.

[15] Kuta, M.—Kitowski, J.: Clustering Polish Texts with Latent Semantic Analysis. In Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J. (Eds.), Proc. of the 10th Int. Conf. on Artificial Intelligence and Soft Computing (ICAISC 2010), Lecture Notes in Artificial Intelligence 6114, Zakopane, Poland, pp. 532–539.

[16] Kuta, M.—Kitowski, J.: Benchmarking High Performance Architectures with Natural Language Processing Algorithms. Computer Science, Vol. 12, 2011, pp. 19–31.

[17] Osinski, S.—Stefanowski, J.—Weiss, D.: Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. In Kłopotek, M., Wierzchon, S., Trojanowski, K. (Eds.), Proc. of the Intelligent Information Processing and Web Mining Conf. 2004, Zakopane, Poland, Advances in Soft Computing, Springer-Verlag, pp. 359–368.

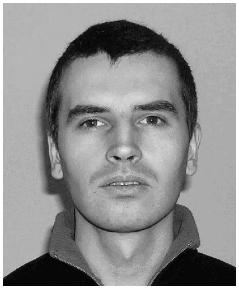
[18] Radovanovic, M.—Ivanovic, M.—Budimac, Z.: Text Categorization and Sorting of Web Search Results. Computing and Informatics, Vol. 28, 2009, No. 6, pp. 861–893.

[19] Weiss, D.: The Corpus of the Polish Daily Rzeczpospolita (Years 1993–2002). `http://www.cs.put.poznan.pl/dweiss/rzeczpospolita` [Accessed July 2012].

[20] Wolinski, M.: Morfeusz – A Practical Tool for the Morphological Analysis of Polish. In Kłopotek, M., Wierzchon, S., Trojanowski, K. (Eds.), Proc. of the Intelligent Information Processing and Web Mining Conf., Advances in Soft Computing, Ustron, Poland 2006, Springer-Verlag, pp. 503–512.

[21] Zhao, Y.—Karypis, G.: Criterion Functions for Document Clustering. Experiments and Analysis. Technical Report 0140, University of Minnesota, Department of Computer Science/Army HPC Research Center Minneapolis 2001.

[22] Zhao, Y.—Karypis, G.: Hierarchical Clustering Algorithms for Document Datasets. Data Mining and Knowledge Discovery, Vol. 10, 2005, No. 2, pp. 141–168.



Marcin Kuta received his M. Sc. in 2001 and Ph. D. in 2011 in computer science from the AGH University of Science and Technology in Kraków (Poland). Since 2003 he has been working at the Institute of Computer Science at the same University. His research interests comprise natural language processing, machine learning techniques, ontology usage, question answering systems and knowledge engineering. He is the author or co-author of 15 scientific papers.

Jacek Kitowski graduated in 1973 from the Electrical Department of the AGH University of Science and Technology in Kraków (Poland). He is the Head of the Computer Systems Group at the Institute of Computer Science of the AGH University of Science and Technology in Kraków, Poland. He became full professor in 2001. He also works for the Academic Computer Centre CYFRONET-AGH, where he is responsible for developing high-performance systems. He is the author or co-author of about 200 scientific papers. His topics of interest include, but are not limited to, large-scale computations, multiprocessor architectures, high availability systems, network computing, Grid services and Grid storage systems, knowledge engineering. He participates in program committees of many conferences, and has been involved in many national and international projects, most notably in EU IST CrossGrid, EU IST Pellucid, EU IST K-WfGrid, EU int.eu.grid and in EU Gredia. He is Polish expert in EU Program Committee "e-Infrastructures" (EU Unit F3 "Research Infrastructures"), Director of PL-Grid Consortium (National Grid Initiative, NGI), member of the Interfaculty Commission of Technical Sciences of the Polish Academy of Arts and Sciences (PAU) and of the Computational Science Section of the Polish Academy of Sciences (PAN), Committee on Informatics.