Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Sparse Distributed Representations for Words with Thresholded Independent Component Analysis

Jaakko J. Väyrynen, Lasse Lindqvist and Timo Honkela

Abstract— **We show that independent component analysis (ICA) can be used to find distributed representations for words that can be further processed by thresholding to produce sparse representations. The applicability of the thresholded ICA representation is compared to singular value decomposition (SVD) in a multiple choice vocabulary task with three data sets.**

I. Introduction

Latent semantic analysis (LSA) [1] is a very popular method for extracting information from text corpora. LSA is based on singular value decomposition (SVD) [2] that removes second order correlations from data. LSA has been shown to produce reasonably low-dimensional latent spaces that can handle various tasks, such as vocabulary tests and essay grading, at human level [1]. The found latent components, however, cannot be understood by humans.

Independent component analysis (ICA, see, e.g., [3] and [4]) can be seen as a whitening followed by a rotation, where the whitening can be produced with SVD. ICA can thus be seen as an extension of LSA. The rotation should transform the latent SVD components into components that are statistically independent of each other, or in the case when the components are not truly independent, it should find "interesting" components. Typical distance measures for LSA are rotation-invariant and would not show differences between ICA and LSA. We are interested in the information encoded by the individual ICA components and how they can be useful.

The ICA has been shown to, e.g., detect topics in document collections (see, e.g., [5] and [6]). Earlier we have shown that the ICA analysis results into meaningful word features (see, [7] and [8]) and that these features correspond to a reasonable extent with categorizations created through human linguistic analysis in [9] and in [10].

In this paper, we present a novel methodological extension. We show that the components found by ICA can be further processed by simple nonlinear filtering methods and produce results with good quality. In particular, the end result is a sparse feature representation of words. We show through practical experiments using three different data sets that this approach exceeds the capacity of the LSA method. An analogical approach can be found from the analysis of natural images, where a soft thresholding of sparse coding is seen as a denoising operator [11].

II. Data

Understanding language requires knowing the different relations between the units of language. Our goal is to find a distributed word representation based on unsupervised learning from actual natural language use. We have a collection of English texts as our source of natural language and our unsupervised learning methods are singular value decomposition and independent components analysis. The representations learned with the methods are applied to a synonym finding task and to an association finding task that measure how well the word representations capture word meanings.

A. Gutenberg Corpus

A collection of 4966 free English e-books were extracted from the Project Gutenberg website [12]. The texts were pruned to exclude poems and the e-book headers and footers were removed. The texts were then concatenated into a single file and preprocessed by removing special characters and replacing numbers by a special symbol and uppercase characters with respective lowercase ones. The final corpus had 319 998 584 tokens (word forms in running texts) and 1 405 298 types (unique word forms). For computational reasons, a subset of the types was selected as the vocabulary to be analyzed.

B. Vocabulary Test Sets

The semantic content of a word representation can be measured with multiple choice vocabulary tests. We chose three test sets. The first one is the synonym part of the TOEFL data set [13] provided by the Institute of Cognitive Science, University of Colorado, Boulder. The second and larger synonym data set was derived from the Moby thesaurus II [14], which is part of the Moby Project. In a synonym test, the task is to choose a synonym or related word from a list of alternatives for a given stem word. The third data set is a word association test derived from the idiosyncratic responses from the free association norms data set [15]. There we defined the task to be select the association produced by a human subject for the same cue word from a list of alternatives. Performance of the methods is measured with precision: the ratio of correct answers to the number of questions in the data set. Questions and the vocabulary were selected to have perfect recall, i.e., all words in the questions were included in the vocabulary.

Jaakko J. Väyrynen is with the Adaptive Informatics Research Centre (AIRC), Helsinki University of Technology, P.O. Box 5400, FI-02015 TKK, Espoo, Finland (phone: +358-9-451-3891; fax: +358-9-451-2711; email: jaakko.j.vayrynen@tkk.fi).

Lasse Lindqvist is with the AIRC (e-mail: llindqvi@cis.hut.fi).

Timo Honkela is with the AIRC (email: timo.honkela@tkk.fi)

1) TOEFL *Synonyms:* In the synonym data set of 80 TOEFL questions, where the task is to select the synonym for the stem word from four alternatives. LSA has been shown to get 64.4 % correct for these questions [1]. Even a level of 97.5 % has been reached by combining several methods, including LSA and an online thesaurus [16]. By comparison, the average score on the 80 questions, for a large sample of applicants to US colleges from non-English speaking countries, is 64.5 % correct [1]. The data set is very limited in size and comparison of methods with only this data set is not sufficient. Also, the baseline precision with guessing from four alternatives is 25 % and chance might play a big role in the precision.

2) Moby Synonyms and Related Words: The Moby Thesaurus of English words and phrases has more than 30 000 entries with 2.5 million synonyms and related terms. We generated multiple choice questions similar to the TOEFL synonym test by selecting a stem from the Moby thesaurus, removing all but one of the synonyms and adding a number of random words from our vocabulary as alternatives. This method allows us to have more questions and alternatives, which makes the test more robust. With four alternatives, the Moby questions scored worse than the TOEFL questions with the methods presented in this paper, which would suggest that, on average, our generated questions are not easier than the hand-made questions. However, when the incorrect alternatives in the TOEFL set were replaced with random words, the precision improved, which means that the hand-made questions are more difficult.

Our vocabulary overlapped with 16 638 entries in the Moby thesaurus and one multiple choice question with 16 alternatives was generated for each one. The baseline precision is 6.25 % with guessing from 16 alternatives.

3) Idiosyncratic Associations: The free association norms data set from the University of South Florida contains idiosyncratic responses, that is, responses given by only one human subject, to more than five thousand cue words. On average, there are approximately 22.15 idiosyncratic responses per cue word with high variation, and typically more idiosyncratic responses are produced than responses given by two or more participants [15].

Similarly to the generated Moby questions, the idiosyncratic association data set was used to generate 4 582 multiple choice questions with 16 alternatives. This data set is smaller than the Moby data set, but still significantly larger than the TOEFL data set.

III. Methods

It has been known already for some time that statistical analysis of the contexts in which a word appears in text can provide reasonable amount of information on the syntactic and semantic roles of the word (see, e.g., [17] and [18]). A typical approach is to calculate a document-term matrix in which the rows correspond to the documents and the columns correspond to the terms. A column is filled with the number of occurrences of the particular term in each document. The similarity of use of any two terms is reflected by the relative similarity of the corresponding two columns in the document-term matrix. Instead of considering the whole documents as contexts, one can also choose the neighboring words, a sentence, a paragraph or some other contextual window. An alternative approach, that is taken here, is to calculate the number of co-occurrences of the particular term with a number of other terms in a contextual window around the analyzed term. This produces a context-term matrix, where each context is defined using terms instead of documents.

A. Contextual Information

Contextual information is a standard way of filtering more dense data from running text. Frequencies of term occurrences, or co-occurrences, in different chunks of texts are typically calculated. The idea behind this is that the relations of words manifest themselves by having related words occur in similar contexts, but not necessary together. Raw contextual data is too sparse for practical use and it has been shown that finding a more compact representation from the raw data can increase the information content by generalizing the data [1].

A context-term matrix $\mathbf{X}$ was calculated using the Gutenberg corpus, where the rows correspond to contexts and the columns represent the terms in the analyzed vocabulary. The context contained frequencies of the 1 000 most common word forms in a 21 word window centered around each occurrence of the analyzed terms. It has been reported that contextual methods are not sensitive to moderate changes in context window size [1] and context window size varies greatly from experiment to experiment [19]. We did not optimize the context window size for our method. The terms included the 50 000 most common word forms in the Gutenberg corpus and additional 29 words that were present in the TOEFL data set but not in the first set so that all of the questions in the TOEFL set could be used. Experiments with only the 1 000 most common terms combined with the TOEFL terms gave similar results. The contextual information was encoded with a bag-of-words model to the matrix $\mathbf{X}$ of size $1\,000 \times 50\,029$. The raw frequency information of the most common words is typically modified using stop-word lists and term weighting, such as the tf·idf method that is suitable for document contexts. We did not use stop-word lists and frequency rank information was preserved by taking the logarithm of the frequencies increased by one, which we have found to be a simple and an efficient method.

B. Singular Value Decomposition

Singular value decomposition learns a latent structure for representing data. Input to singular value decomposition is a $m \times n$ matrix $\mathbf{X}$. The SVD method finds the decomposition $\mathbf{X} = \mathbf{UDV}^T$, where $\mathbf{U}$ is an $m \times r$ matrix of left singular vectors from the standard eigenvectors of square symmetric matrix $\mathbf{XX}^T$, $\mathbf{V}$ is an $n \times r$ matrix of right singular vectors from the eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{D}$ is a diagonal $r \times r$ matrix whose non-zero values are the square roots of the eigenvalues of $\mathbf{XX}^T$ or (equivalently) $\mathbf{X}^T\mathbf{X}$, and $r = \min(n, m)$ is the rank of $\mathbf{X}$. A lossy dimension reduction to $l \leq r$ components can be achieved by discarding small eigenvalues.

In latent semantic analysis, that is based on SVD, the input matrix $\mathbf{X}$ is a context-term matrix representing the weighted frequencies of terms in text passages or other contexts. The method can handle tens of thousands of terms and contexts. Dimension is typically lowered to a few hundred components, that reduces noise and generalizes the data by finding a latent semantic representation for words. Words and texts can be compared by their respective vectorial representations in the latent space. We calculated SVD with the PROPACK [20] package for Matlab.

C. Independent Component Analysis

Independent component analysis uses higher-order statistics compared to singular value decomposition that only removes second-order correlations. ICA finds a decomposition $\mathbf{Z} = \mathbf{BS}$ for a data matrix $\mathbf{Z}$, where $\mathbf{B}$ is a mixing matrix of weights for the independent components in the rows of matrix $\mathbf{S}$. The task is usually to find a separating matrix $\mathbf{W} = \mathbf{B}^{-1}$ that produces independent components $\mathbf{S} = \mathbf{WZ}$.

If data $\mathbf{Z}$ is white, i.e., the covariance matrix is an identity matrix, it suffices to find a rotation that produces maximally independent components [4]. The right singular values $\mathbf{V}$ produced by SVD are white and thus SVD can be seen as a preprocessing step to ICA. This is illustrated in Figure 1. The ICA rotation should find components that are more interesting and structure the semantic space in a meaningful manner. We calculated ICA with the FastICA package for Matlab [21].

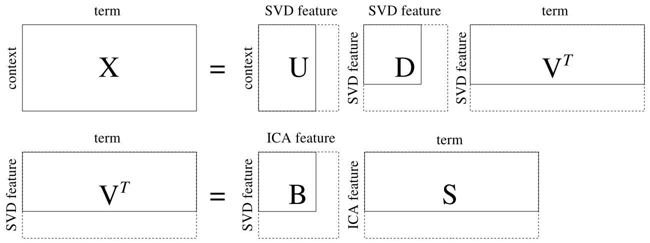

Fig. 1. ICA can be represented as an extension of SVD, where the white SVD components $\mathbf{Z} = \mathbf{V}^T$ are transformed with a rotation matrix $\mathbf{B}$ to find the ICA components $\mathbf{S}$. SVD is approximated for a reduced dimension from the original dimension of the data matrix $\mathbf{X}$, marked here with the solid and dashed lines, respectively.

D. Word Space and Thresholding

The orthographic representation of words does not give a direct way of comparing the similarity of words. The vectorial representations of the raw contextual data, the SVD representation and the ICA representation, however, represent words as points in space. The locations of the words in the space are a result of the contexts used for collecting the raw contextual data, text occurrences of the words in the corpus, and the components found by LSA and ICA, respectively.

Two words in the space are compared through their vectorial representations $\mathbf{a}$ and $\mathbf{b}$, which are the respective two columns in the matrix $\mathbf{X}$, $\mathbf{V}^T$ or $\mathbf{S}$. We chose the cosine measure

$$d(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{1}$$

that gives the cosine of the angle between the two vectors $\mathbf{a}$ and $\mathbf{b}$. In a multiple choice vocabulary test, the stem word is compared to all alternatives and the closest word according to (1) is chosen as the answer. The measure works similarly for word vectors with thresholded values. The measure is a typical one in LSA and is invariant to the rotation found by ICA. This ensures that the differences between the methods are due to the thresholding.

SVD has an inbuilt method of selection of the components in descending magnitude of the variances in the directions of the eigenvectors. This is an efficient way to reduce dimension and optimal in mean square error sense.

Similar treatment does not work with ICA that considers the components to have unit variance and does not order the components. Instead, the dimension is already reduced with SVD and followed by an ICA rotation. We further processed the ICA space by selecting a number of active components for each word vector. The rest of the components are selected to be inactive. The selected components depend on the word, but the same number of components was selected for each word. As the mean was not removed from the data, the component values selected as inactive were set to the mean component value. The number of selected active components for all word vectors was the same, which makes the comparison between ICA thresholding and SVD feasible.

The selection of the active components for each word vector was based on ordering normalized, i.e., zero mean and unit variance, absolute component values of the word vectors. Low values closer to the mean component value are thought as less active than values that diverge from the mean. Our assumption is that the ICA components are linguistically more separated than the latent SVD components and that only a few active components are needed to represent each word. The inactive ICA component values would thus be less important for that particular word and can be thresholded. We did not want to fix the thresholding parameter, i.e., the fixed number of components that are thresholded for each word vector. Instead, we varied this parameter and show results with thresholding from the full model with no thresholding to the minimal model with only one retained component for each word.

Dimension reduction with SVD can be thought as thresholding the same components for each word vector. This has the additional advantage that the inactive components can be dropped out and the dimension can actually be reduced, whereas ICA thresholding only makes the representation more sparse without dimension reduction. Thresholding with SVD was also tested for comparison.

IV. Results

Singular value decomposition orders the latent components according to the eigenvalues of the covariance matrix. This allows a natural and efficient way of reducing dimension. In this paper, the necessary number of components for efficient working of the latent space as a semantic representation is measured with vocabulary tests. Figure 2 shows precision of the SVD space with the Moby questions w.r.t. the number of components. The peak precision is seen with approximately 80 components. This can be explained by many noise components with small eigenvalues that do not contribute positively to the precision of the classification system. The peak value can be thought to be an optimal number of components and a good starting point for ICA. The 95 % confidence intervals for the Moby data set would overlap the precision curve and are not shown in this section.

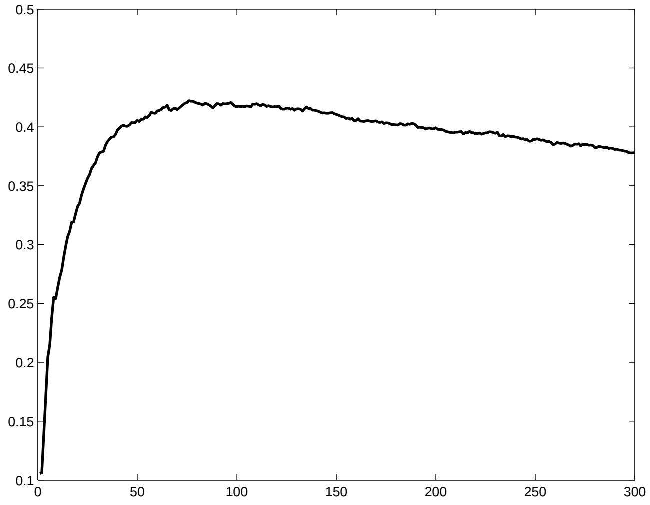

Fig. 2. Precision of the SVD space with the Moby data set w.r.t. dimension of the latent space. With more than 80 dimensions the precision actually drops from the peak value.

Independent component analysis does not order the components and feature selection must be done by other means. Dimension reduction and whitening with SVD is the standard practice for ICA, but this would only add a rotation of the space compared to SVD. The distance measure in (1) is rotation-invariant and basic ICA would not contribute significant changes to the classification result. The assumption that the ICA components are meaningful as such suggests that the inactive components for words can be de-selected by thresholding the components with values close to the mean component value (see, e.g., [22]). This makes the word representation sparse and could remove noise. Dimension reduction in preprocessing is important, as it will remove noise, as ICA does not order the components and considers each component to have unit variance. Figure 3 plots the precision of ICA with thresholding starting from 20, 40 and 80 components for the Moby data set w.r.t. to the number of active components. The SVD precision in Figure 2 is shown as reference. It can be seen that precision grows faster with ICA and levels out to the SVD precision when no components are thresholded.

Thresholding with SVD does not improve the precision, as can be seen in Figure 4, that compares the precisions of SVD and SVD with thresholding starting with 80 components with the Moby data set. SVD with thresholding performs similarly to the basic SVD. With a very small number of starting components, ICA and SVD thresholding methods perform quite similarly. This is understandable, since if the dimension of the representation is too small, it may not be possible to separate semantic components from each other. With an optimal number of starting components, selected with the peak SVD precision, ICA with thresholding is more accurate and outperforms SVD with all thresholding values. If the starting dimension is too high and includes noise components, however, ICA with thresholding does not reach the same peak level as SVD.

Similar tests were run with the questions derived from

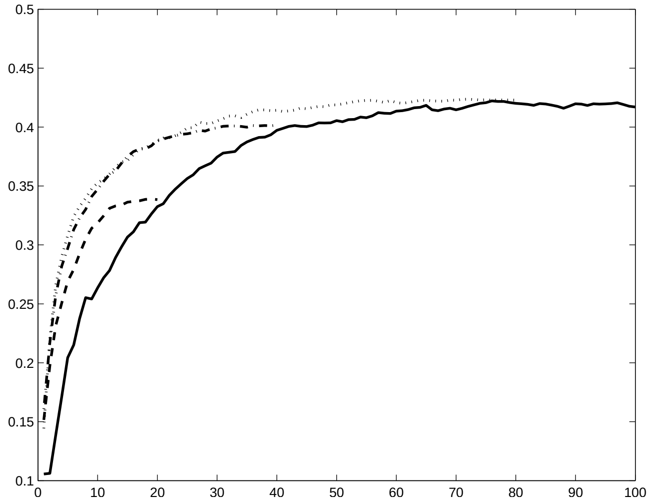

Fig. 3. Precision of ICA with thresholding with the Moby data set w.r.t. the number of active components. ICA word representations are thresholded to include only a selected number of components from 20 (dashed), 40 (dash dotted) and 80 (dotted) components. The SVD precision (solid) up to 100 components is given as reference.

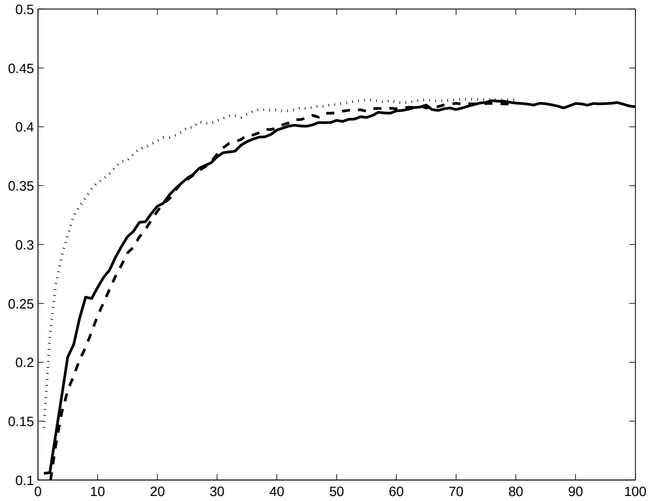

Fig. 4. Precisions of the SVD (solid), SVD with thresholding with 80 components (dashed) and ICA with thresholding with 80 components (dotted) with the Moby data set w.r.t. the number of active components.

the idiosyncratic association data set and results are quite similar, even the precision is approximately at the same level. Figure 5 shows the result with 60 components for ICA and SVD with thresholding and comparison to SVD precision. Similarly to the Moby questions, the precision did not grow after the peak value approximately at 100 components.

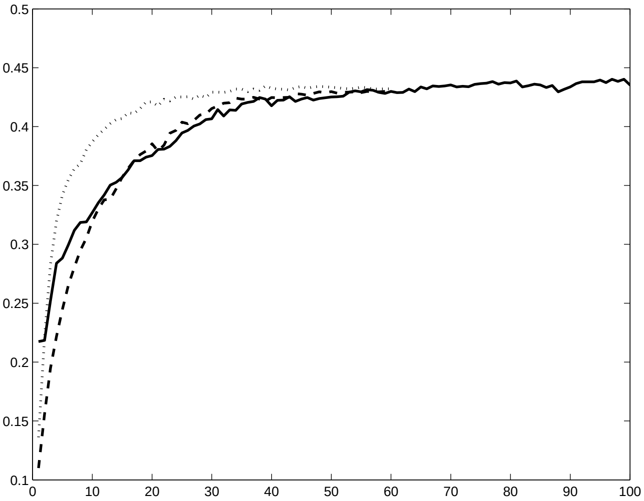

Fig. 5. Precisions of the SVD (solid), SVD with thresholding with 60 components (dashed) and ICA with thresholding with 60 components (dotted) with the idiosyncratic association data set w.r.t. the number of active components.

Results for the TOEFL data set, shown in Figure 6, show that here SVD with thresholding performs equally well to ICA with thresholding, both with 106 components. The hand-picked alternatives for the TOEFL questions make the alternatives for each question to be close to each other. This seems to affect the precision as SVD with thresholding works equally well as ICA with thresholding. Changing the hand-selected incorrect alternatives to random words produces results similar to Figure 4 and Figure 5, where SVD with thresholding does not perform better than SVD. This suggests that ICA with thresholding performs equally well for all words whereas SVD with thresholding works only with words that are all very similar to each other.

An interesting result is the highest precision, 81.25 %, that was reached with 106 components with SVD for the TOEFL data set. That precision is equal to the result obtained with document retrieval with a window of 16 words, point-wise mutual information and a 53 billion word web corpus [19]. Comparison to the LSA result of 64.4 % with 60 000 words, 30 000 document contexts and dimension reduction to 300 with SVD [1], however, shows a huge improvement. The main difference in our experiments is the use of co-occurrences of terms in window contexts instead of the document-based approach. Thus, it seems that the selection of context type has a crucial effect on the results.

V. Conclusions

In this paper, distributed semantic feature representations for words are extracted using contextual information from natural language data. Especially, we demonstrate how independent component analysis finds an explicit representation for words, compared to the latent representation found by singular value decomposition. This is carried out by additional nonlinear filtering, which preserves the semantically rich component values for each word, and thresholds the less important components the mean component values.

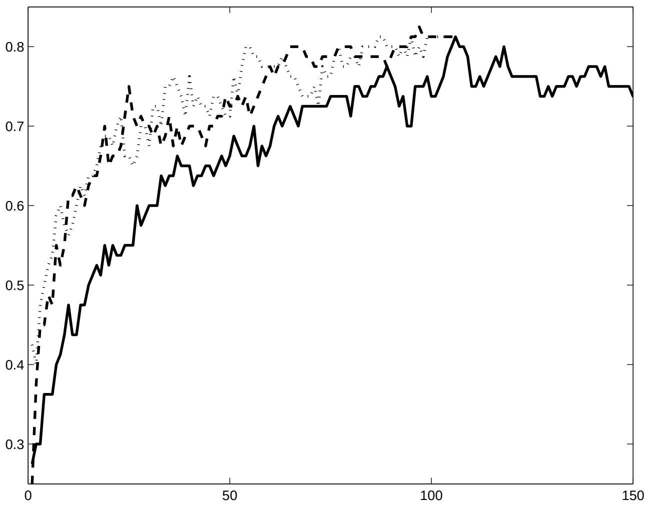

Fig. 6. Precisions of the SVD (solid), SVD with thresholding with 106 components (dashed) and ICA with thresholding with 106 components (dotted) with the TOEFL data set w.r.t. the number of active components.

The semantic information content of the different representations is measured with multiple choice vocabulary tests, in which the locations of the words in the space represents their relationship to each other. Two alternatives are proposed to complement the traditional synonym TOEFL data set. The first one is a synonym questions data set generated from the electronic Moby thesaurus. The second data set is generated from idiosyncratic associations from human subjects. The proposed data sets are much larger and the number of alternatives can easily be extended, but on the other hand, there is no knowledge of human level performance.

Independent component analysis and singular value decomposition are both examined as methods for extracting word representations from raw contextual information. An optimal dimension can be seen to contain all semantic information and to exclude noise. Additionally, ICA can be seen to find such a rotation of the representation that the components reflect more meaningful concepts. Thresholding of inactive values with ICA gives a sparse representation for words with much less degradation of precision in vocabulary tests than SVD with or without thresholding. SVD with thresholding performs equally well to ICA with thresholding when considering words that are close to each other, but unlike ICA with thresholding, it does not generalize to all words.

The parameters and preprocessing were chosen to represent an LSA approach, in order to avoid tuning the method to favor ICA. This is reflected in the fact that we were able to reach a precision of 81.25 % with full SVD for the TOEFL synonym questions, that did equally well to the best single reported method but does not reach the level of a combination of different methods.

The results shown in this article indicate that it is possible to create automatically a sparse representation for words. Moreover, the emergent features in this representation seem to correspond with some linguistically relevant features. When the context is suitably selected for the ICA analysis, the emergent features mostly correspond to some semantic selection criteria. Traditionally, linguistic features have been determined manually. For instance, case grammar is a classical theory of grammatical analysis [23] that proposes to analyze sentences as constituted by the combination of a verb plus a set of deep cases, i.e., semantic roles. Numerous different theories and grammar formalisms exist that provide a variety of semantic or syntactic categories into which words need to be manually classified.

Statistical methods such as SVD and ICA are able to analyze context-term matrices to produce automatically useful representations. ICA has the additional advantage, especially when combined with some additional processing steps reported in this article, over SVD (and thus LSA) that the resulting representation is sparse and each component of the representation is meaningful as such. As the LSA method is already very popular, we assume that the additional advantages brought by this method will further strengthen the movement from manual analysis to an automated analysis.

Acknowledgment

This work was supported in part by the Helsinki Graduate School of Computer Science and Engineering, the Academy of Finland through the Finnish Centre of Excellence Programme and Nokia Foundation.

References

[1] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, pp. 211–240, 1997.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *J. American Sociecty of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] P. Comon, "Independent Component Analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[4] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.

[5] C. L. Isbell, Jr. and P. Viola, "Restructuring sparse high dimensional data for effective retrieval," in *Proc. Conf. on Advances in Neural Information Processing Systems (NIPS 1998)*, 1999, pp. 480–486.

[6] E. Bingham, A. Kabán, and M. Girolami, "Finding topics in dynamical text: application to chat line discussions," in *Poster Proc. 10th Intern. World Wide Web Conf. (WWW'10)*, 2001, pp. 198–199.

[7] T. Honkela and A. Hyvärinen, "Linguistic feature extraction using independent component analysis," in *Proc. Intern. Joint Conf. on Neural Networks (IJCNN)*, 2004, pp. 279–284.

[8] T. Honkela, A. Hyvärinen, and J. Väyrynen, "Emergence of linguistic features: Independent component analysis of contexts," in *Proc. 9th Neural Computation and Psychology Workshop (NCPW9): Modeling Language Cognition and Action*, 2004, pp. 129–138.

[9] J. J. Väyrynen, T. Honkela, and A. Hyvärinen, "Independent component analysis of word contexts and comparison with traditional categories," in *Proc. 6th Nordic Signal Processing Symposium (NORSIG 2004)*, 2004, pp. 300–303.

[10] J. J. Väyrynen and T. Honkela, "Comparison of independent component analysis and singular value decomposition in word context analysis," in *Proc. Intern. and Interdisciplinary Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, 2005, pp. 135–140.

[11] E. Oja, A. Hyvärinen, and P. Hoyer, "Image feature extraction and denoising by sparse coding," *Pattern Analysis & Applications*, vol. 2, no. 2, pp. 104–110, 1999.

[12] Project Gutenberg. [Online]. Available: http://www.gutenberg.org

[13] Educational Testing Service. Test of English as foreign language. [Online]. Available: http://www.ets.org

[14] G. Ward. (1993) Moby thesaurus II. [Online]. Available: http://www.dcs.shef.ac.uk/research/ilash/Moby/

[15] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber. (1998) The University of South Florida word association, rhyme, and word fragment norms. [Online]. Available: http://www.usf.edu/FreeAssociation/

[16] P. D. Turney, M. L. Littman, J. Bigham, and V. Schnayder, "Combining independent modules to solve multiple-choice synonym and analogy problems," in *Proc. Intern. Conf. on Recent Advances in Natural Language Processing (RANLP-03)*, 2003, pp. 482–489.

[17] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[18] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biological Cybernetics*, vol. 61, no. 4, pp. 241–254, 1989.

[19] E. Terra and C. L. A. Clarke, "Frequency estimates for statistical word similarity measures," in *Proc. Human Language Technology Conf. and Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2003, pp. 165–172.

[20] R. M. Larsen. (2004) PROPACK: A software package for the symmetric eigenvalue problem and singular value problems on Lanczos and Lanczos bidiagonalization with partial reorthogonalization. SCCM, Stanford University. [Online]. Available: http://soi.stanford.edu/˜rmunk/PROPACK/

[21] The FastICA Team. (2005) The FastICA MATLAB package. Helsinki University of Technology. [Online]. Available: http://www.cis.hut.fi/projects/ica/fastica/

[22] B. Tang, M. Shepherd, E. Milios, and M. I. Heywood, "Comparing and combining dimension reduction techniques for efficient text clustering," in *Proc. Intern. Workshop on Feature Selection for Data Mining (FSDM 2005): Interfacing Machine Learning and Statistics*, 2005, pp. 17–26.

[23] C. J. Fillmore, *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, 1968, ch. The Case for Case, pp. 1–88.