Cross-Lingual Word Representations via Spectral Graph Embeddings

Takamasa Oshikiri, Kazuki Fukui, Hidetoshi Shimodaira
Division of Mathematical Science, Graduate School of Engineering Science
Osaka University, Japan
1-3 Machikaneyama-cho, Toyonaka, Osaka
{oshikiri, fukui, shimo}@sigmath.es.osaka-u.ac.jp

Abstract

Cross-lingual word embeddings are used for cross-lingual information retrieval or domain adaptations. In this paper, we extend Eigenwords, spectral monolingual word embeddings based on canonical correlation analysis (CCA), to cross-lingual settings with sentence-alignment. For incorporating cross-lingual information, CCA is replaced with its generalization based on the spectral graph embeddings. The proposed method, which we refer to as Cross-Lingual Eigenwords (CL-Eigenwords), is fast and scalable for computing distributed representations of words via eigenvalue decomposition. Numerical experiments of English-Spanish word translation tasks show that CL-Eigenwords is competitive with state-of-the-art cross-lingual word embedding methods.

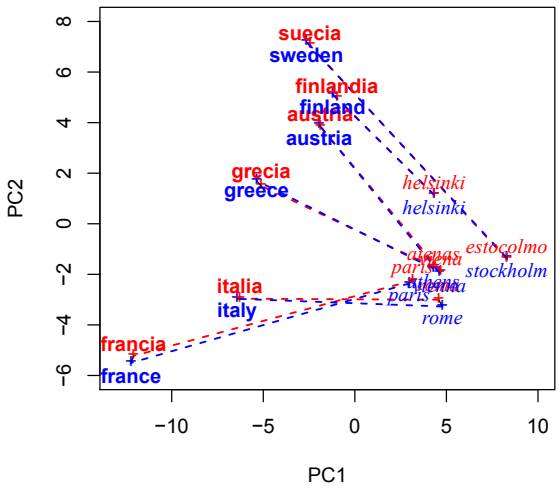

Figure 1: PCA projections (PC1 and PC2) of CL-Eigenwords of countries (bold) and its capitals (italic) in Spanish (red) and English (blue). Word vectors of the two languages match quite well, although they are computed using sentence-level alignment without knowing word-level alignment. 100-dim word representations are used for PCA computation.

1 Introduction

There have been many methods proposed for word embeddings. Neural network based models are popular, and one of the most major approaches is the skip-gram model (Mikolov et al., 2013b), and some extended methods have also been developed (Levy and Goldberg, 2014a; Lazaridou et al., 2015). The skip-gram model has many interesting syntactic and semantic properties, and it can be seen as the factorization of a word-context matrix whose elements represent pointwise mutual information (Levy and Goldberg, 2014b). However, word embeddings based on neural networks (without neat implementation) can be very slow in general, and it is sometimes difficult to understand how they work. Recently, a simple spectral method, called Eigenwords, for word embeddings is proposed (Dhillon et al., 2012; Dhillon et al., 2015). It is based on canonical correlation analysis (CCA) for computing word vectors by maximizing correlations between words and their contexts. Eigenword algorithms are fast and scalable, yet giving good performance comparable to neural network approaches for capturing the meaning of words from their context.

The skip-gram model, originally proposed for monolingual corpora, has been extended to cross-lingual settings. Given two vector representations of two languages, a linear transformation between the two spaces is trained from a set of word pairs for translation task (Mikolov et al., 2013a), while other researchers use CCA for learning linear projections to a common vector space where

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 493–498,
Berlin, Germany, August 7-12, 2016.

translation pairs are strongly correlated (Faruqui and Dyer, 2014). These methods require word-alignment in the training data, while some multilingual corpora have only coarse information such as a set of sentence pairs or paragraph pairs. Recently, extensions of the skip-gram model requiring only sentence-alignment have been developed by introducing cross-lingual losses in the objective of the original models (Gouws et al., 2015; Coulmance et al., 2015; Shi et al., 2015).

In this paper, instead of the skip-gram model, we extend Eigenwords (Dhillon et al., 2015) to cross-lingual settings with sentence-alignment. Our main idea is to replace CCA, which is applicable to only two different kinds of data, with a generalized method (Nori et al., 2012; Shimodaira, 2016) based on spectral graph embeddings (Yan et al., 2007) so that the Eigenwords can deal with two or more languages for cross-lingual word embeddings. Our proposed method, referred to as Cross-Lingual Eigenwords (CL-Eigenwords), requires only sentence-alignment for capturing cross-lingual relationships. The method is very simple in mathematics as well as computation; it involves a generalized eigenvalue problem, which can be solved by fast and scalable algorithms such as the randomized eigenvalue decomposition (Halko et al., 2011).

Fig. 1 shows an illustrative example of cross-lingual word vectors obtained by CL-Eigenwords. Although only sentence-alignment is available in the corpus, word-level translation is automatically captured in the vector representations; the same words (countries and capitals) in the two languages are placed in close proximity to each other; *greece* is close to *grecia* and *rome* is close to *roma*. In addition, the same kinds of relationships between word pairs share similar directions in the vector space; the direction from *sweden* to *stockholm* is nearly parallel to the direction from *finland* to *helsinki*.

We evaluate the word vectors obtained by our method on the English-Spanish cross-lingual translation task and compare the results with those of state-of-the-art methods, showing that our proposed method is competitive with those existing methods. We use Europarl corpus for learning the vector representation of words. Although the experiments in this paper are conducted using bilingual corpus, our method can be easily applied to three or more languages.

2 Eigenwords (One Step CCA)

CCA (Hotelling, 1936) is a multivariate analysis method for finding optimal projections of two sets of data vectors by maximizing the correlations. Applying CCA to pairs of raw word vector and raw context vector, Eigenword algorithms attempt to find low dimensional vector representations of words (Dhillon et al., 2012). Here we explain the simplest version of Eigenwords called One Step CCA (OSCCA).

We have monolingual corpus consisting of T tokens; $(t_i)_{i=1,\ldots,T}$, and the vocabulary consisting of V word types; $\{v_i\}_{i=1,\ldots,V}$. Each token t_i is drawn from this vocabulary. We define word matrix $\mathbf{V} \in \{0,1\}^{T\times V}$ whose i-th row encodes token t_i by 1-of-V representation; the j-th element is 1 if the word type of t_i is v_j, 0 otherwise.

Let h be the size of context window. We define context matrix $\mathbf{C} \in \{0,1\}^{T\times 2hV}$ whose i-th row represents the surrounding context of token t_i with concatenated 1-of-V encoded vectors of $(t_{i-h},\ldots,t_{i-1},t_{i+1},\ldots,t_{i+h})$.

We apply CCA to T pairs of row vectors of $\mathbf{V}$ and $\mathbf{C}$. The objective function of CCA is constructed using $\mathbf{V}^\top\mathbf{V}$, $\mathbf{V}^\top\mathbf{C}$, $\mathbf{C}^\top\mathbf{C}$ which represent occurrence and co-occurrence counts of words and contexts. In Eigenwords, however, we use $\boldsymbol{\mathcal{C}}_{VV} \in \mathbb{R}_+^{V\times V}$, $\boldsymbol{\mathcal{C}}_{VC} \in \mathbb{R}_+^{V\times 2hV}$, $\boldsymbol{\mathcal{C}}_{CC} \in \mathbb{R}_+^{2hV\times 2hV}$ with the following preprocessing of these matrices before constructing the objective function. First, centering-process of $\mathbf{V}$ and $\mathbf{C}$ is omitted, and off-diagonal elements of $\mathbf{C}^\top\mathbf{C}$ are ignored for simplifying the computation of inverse matrices. Second, we take the square root of the elements of these matrices for "squashing" the heavy-tailed word count distributions. Finally, we obtain vector representations of words as $\boldsymbol{\mathcal{C}}_{VV}^{-1/2}(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K)$, where $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K \in \mathbb{R}^V$ are left singular vectors of $\boldsymbol{\mathcal{C}}_{VV}^{-1/2}\,\boldsymbol{\mathcal{C}}_{VC}\,\boldsymbol{\mathcal{C}}_{CC}^{-1/2}$ corresponding to the K largest singular values. The computation of SVD is fast and scalable using recent idea of random projections (Halko et al., 2011).

3 Cross-Lingual Eigenwords

In this section, we introduce **Cross-Lingual Eigenwords (CL-Eigenwords)**, a novel method for cross-lingual word embeddings. Suppose that we have parallel corpora that contain L languages. Schematic diagrams of Eigenwords and

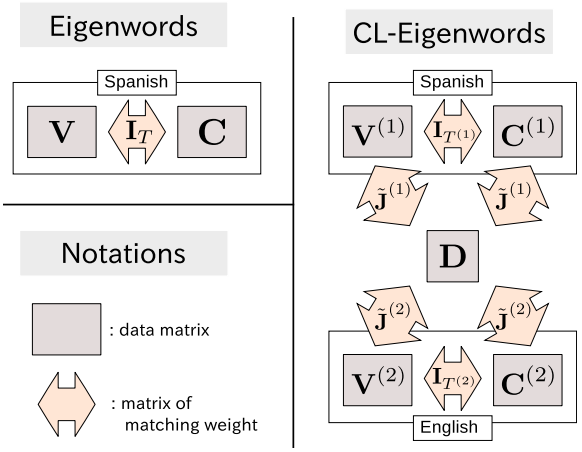

Figure 2: Eigenwords are CCA-based spectral monolingual word embeddings. CL-Eigenwords are CDMCA-based spectral cross-lingual word embeddings, where the two (or more) languages are linked by sentence-alignment.

CL-Eigenwords (with $L = 2$) are shown in Fig. 2.

In the same way as the monolingual Eigenwords, we denote the word matrix and the context matrix for ℓ-th language by $\mathbf{V}^{(\ell)} \in \mathbb{R}_+^{T^{(\ell)} \times V^{(\ell)}}$ and $\mathbf{C}^{(\ell)} \in \mathbb{R}_+^{T^{(\ell)} \times 2h^{(\ell)} V^{(\ell)}}$ respectively, where $V^{(\ell)}$ is the size of vocabulary, $T^{(\ell)}$ is the number of tokens, and $h^{(\ell)}$ is the size of context window. There are D sentences (or paragraphs) in the multilingual corpora, and each token is included in one of the sentences. The sentence-alignment is represented in the matrix $\mathbf{J}^{(\ell)} \in \mathbb{R}_+^{T^{(\ell)} \times D}$ whose (i, j)-element $J_{i,j}^{(\ell)}$ is set to 1 if the i-th token $t_i^{(\ell)}$ of ℓ-th language corpus comes from the j-th sentence or 0 otherwise. We also define document matrix $\mathbf{D}$ whose j-th row encodes j-th sentence by 1-of-D representation; $\mathbf{D} = \mathbf{I}_D$, where $\mathbf{I}_D$ represents D-dimensional identity matrix.

The goal of CL-Eigenwords is to construct vector representations of words of two (or more) languages from multilingual corpora at the same time. This problem is formulated as an example of Cross-Domain Matching Correlation Analysis (CDMCA) (Shimodaira, 2016), which deals with many-to-many relationships between data vectors from multiple sources. CDMCA is based on the spectral graph embeddings (Yan et al., 2007), and attempts to find optimal linear projections of data vectors so that associated transformed vectors are placed in close proximity to each other. The strength of association between two vectors is specified by a nonnegative real value called *matching weight*. Since CDMCA includes CCA and a variant of Latent Semantic Indexing (LSI) (Deerwester et al., 1990) as special cases, CL-Eigenwords can be interpreted as LSI-equipped Eigenwords (See Appendix).

In CL-Eigenwords, the data vectors are given as $\boldsymbol{v}_i^{(\ell)}, \boldsymbol{c}_i^{(\ell)}, \boldsymbol{d}_i$, namely, the i-th row vectors of $\mathbf{V}^{(\ell)}, \mathbf{C}^{(\ell)}, \mathbf{D}$, respectively. The matching weights between row vectors of $\mathbf{V}^{(\ell)}$ and $\mathbf{C}^{(\ell)}$ are specified by the identity matrix $\mathbf{I}_{T^{(\ell)}}$ because the data vectors are in one-to-one correspondence. On the other hand, the matching weights between row vectors of $\mathbf{V}^{(\ell)}$ and $\mathbf{D}$ as well as those between $\mathbf{C}^{(\ell)}$ and $\mathbf{D}$ are specified by $\tilde{\mathbf{J}}^{(\ell)} = b^{(\ell)} \mathbf{J}^{(\ell)}$, the sentence-alignment matrix multiplied by a constant $b^{(\ell)}$. Then we will find linear transformation matrices $\mathbf{A}_V^{(\ell)}, \mathbf{A}_C^{(\ell)}, \mathbf{A}_D$, $(\ell = 1, 2, \ldots, L)$ to K-dimensional vector space by minimizing the objective function

$$\begin{aligned}
&\sum_{\ell=1}^{L} \sum_{i=1}^{T^{(\ell)}} \| \boldsymbol{v}_i^{(\ell)} \mathbf{A}_V^{(\ell)} - \boldsymbol{c}_i^{(\ell)} \mathbf{A}_C^{(\ell)} \|_2^2 \\
&+ \sum_{\ell=1}^{L} \sum_{i=1}^{T^{(\ell)}} \sum_{j=1}^{D} \tilde{J}_{i,j}^{(\ell)} \| \boldsymbol{v}_i^{(\ell)} \mathbf{A}_V^{(\ell)} - \boldsymbol{d}_j \mathbf{A}_D \|_2^2 \\
&+ \sum_{\ell=1}^{L} \sum_{i=1}^{T^{(\ell)}} \sum_{j=1}^{D} \tilde{J}_{i,j}^{(\ell)} \| \boldsymbol{c}_i^{(\ell)} \mathbf{A}_C^{(\ell)} - \boldsymbol{d}_j \mathbf{A}_D \|_2^2 \qquad (1)
\end{aligned}$$

with a scale constraint for projection matrices. Note that the first term in (1) is equivalent to that of CCA between words and contexts, namely the objective of monolingual Eigenwords, and therefore word vectors of two languages are obtained as row vectors of $\mathbf{A}_V^{(\ell)}$ $(\ell = 1, 2, \ldots, L)$.

Hereafter, we assume $L = 2$ for notational simplicity. A generalization to the case $L > 2$ is straightforward; redefine $\mathbf{X}$, $\mathbf{W}$, $\mathbf{A}$ below by repeating the submatrices, such as $\mathbf{V}^{(\ell)}$ and $\mathbf{C}^{(\ell)}$, for L times. For solving the optimization problem, we define

$$\mathbf{X} = \begin{pmatrix} \mathbf{V}^{(1)} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{C}^{(1)} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{V}^{(2)} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{C}^{(2)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{D} \end{pmatrix},$$

$$\mathbf{W} = \begin{pmatrix} \mathbf{O} & \mathbf{I}_{T^{(1)}} & \mathbf{O} & \mathbf{O} & \tilde{\mathbf{J}}^{(1)} \\ \mathbf{I}_{T^{(1)}} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \tilde{\mathbf{J}}^{(1)} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}_{T^{(2)}} & \tilde{\mathbf{J}}^{(2)} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}_{T^{(2)}} & \mathbf{O} & \tilde{\mathbf{J}}^{(2)} \\ \tilde{\mathbf{J}}^{(1)\top} & \tilde{\mathbf{J}}^{(1)\top} & \tilde{\mathbf{J}}^{(2)\top} & \tilde{\mathbf{J}}^{(2)\top} & \mathbf{O} \end{pmatrix},$$

$$\mathbf{A}^\top = (\mathbf{A}_V^{(1)\top}, \mathbf{A}_C^{(1)\top}, \mathbf{A}_V^{(2)\top}, \mathbf{A}_C^{(2)\top}, \mathbf{A}_D^\top).$$

Method	Time [min]	1 – 1000 es → en P@1	P@5	1 – 1000 en → es P@1	P@5	5001 – 6000 es → en P@1	P@5	5001 – 6000 en → es P@1	P@5
Edit distance	-	29.1	37.8	20.6	34.4	28.5	40.0	26.4	33.5
BilBOWA (40 dim.)	* 4.6	46.7	59.6	43.6	56.4	44.6	53.6	49.4	**58.7**
BilBOWA (100 dim.)	* 7.5	43.3	55.9	36.8	49.0	43.6	53.3	48.6	57.9
BilBOWA (200 dim.)	* 11.6	38.8	52.2	29.7	43.2	43.3	52.0	47.3	57.2
CL-LSI (40 dim.)	1.4	45.9	54.8	46.9	55.8	31.6	38.5	40.7	45.1
CL-LSI (100 dim.)	2.4	51.7	62.9	48.5	61.8	41.6	49.8	42.8	49.1
CL-LSI (200 dim.)	5.1	55.2	66.5	50.7	65.5	45.5	54.7	45.6	51.9
CL-Eigenwords (40 dim.)	9.5	54.7	66.2	53.3	65.7	40.3	49.2	44.7	50.0
CL-Eigenwords (100 dim.)	19.6	57.7	71.3	54.9	70.3	47.9	59.0	49.3	54.6
CL-Eigenwords (200 dim.)	37.5	**58.7**	**72.4**	**56.2**	**72.2**	**51.6**	**62.4**	**50.6**	55.7

Table 1: Computational times (in minutes) and word translation accuracies (in percent, higher is better) evaluated by Precision@n using the 1,000 test words (the 1st to 1,000th most frequent words or the 5,001st to 6,000th most frequent words). Shown are for Spanish (es) to English (en) translation and for English (en) to Spanish (es) translation. * BilBOWA is executed on 3 threads, while CL-LSI and CL-Eigenwords are executed on a single thread.

Also define $\mathbf{H} = \mathbf{X}^\top \mathbf{W} \mathbf{X}$, $\mathbf{G} = \mathbf{X}^\top \mathbf{M} \mathbf{X}$, $\mathbf{M} = \mathrm{diag}(\mathbf{W}\mathbf{1})$. Then the optimization problem (1) is equivalent to maximizing $\mathrm{Tr}(\mathbf{A}^\top \mathbf{H} \mathbf{A})$ with a scale constraint $\mathbf{A}^\top \mathbf{G} \mathbf{A} = \mathbf{I}_K$. Following the Eigenwords implementation (Dhillon et al., 2015), we replace $\mathbf{H}, \mathbf{G}$ with $\boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{G}}$ by ignoring the non-diagonal elements of $\mathbf{G}$ and taking the square root of elements in $\mathbf{H}, \mathbf{G}$. The optimization problem is solved as a generalized eigenvalue problem, and the word representations, as well as those for contexts and sentences, are obtained as row vectors of $\hat{\mathbf{A}} = \boldsymbol{\mathcal{G}}^{-1/2}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)$, where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ are eigenvectors of $(\boldsymbol{\mathcal{G}}^{-1/2})^\top \boldsymbol{\mathcal{H}} \boldsymbol{\mathcal{G}}^{-1/2}$ for the K largest eigenvalues. We choose K so that all the K eigenvalues are positive. As in the case of monolingual Eigenwords, we can exploit fast implementations such as the randomized eigenvalue decomposition (Halko et al., 2011); our computation in the experiments is only approximation based on the low-rank factorization with rank $2K$.

For measuring similarities between two word vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^K$, we use the weighted cosine similarity

$$\mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) = (\|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2)^{-1} \sum_{i=1}^{K} \lambda_i x_i y_i,$$

where λ_i is the i-th largest eigenvalue.

4 Experiments

The implementation of our method is available on GitHub[1]. Following the previous works (Mikolov et al., 2013a; Gouws et al., 2015), we use only the first 500K lines of English-Spanish sentence-aligned parallel corpus of Europarl (Koehn, 2005) for numerical experiments.

4.1 Word Translation Tasks

Experiments are performed in similar settings as the previous works based on the skip-gram model (Mikolov et al., 2013a; Gouws et al., 2015). We extract 1,000 test words with frequency rank 1–1000 or 5001–6000 from the source language, and translate these words to the target language using Google Translate, assuming they are the correct translations. Then, we evaluate the translation accuracies of each method with precision@n as the fraction of correct translations for the test words being in the top-n words of the target language returned by each method.

4.2 Baseline Systems

We compare CL-Eigenwords with the following three methods.

Edit distance Finding the nearest words measured by Levenshtein distance.

CL-LSI Cross-Language LSI (CL-LSI) (Littman et al., 1998) is not originally for word embeddings. However, since this method can be used for cross-lingual information retrieval, we select it as one of our baselines. For each language, we construct the term-document matrix of size $V^{(\ell)} \times D$ whose (i, j)-element represents the frequency of i-th word in j-th sentence. Then LSI is applied to the concatenated matrix of size $(V^{(1)} + V^{(2)}) \times D$.

BilBOWA BilBOWA (Gouws et al., 2015) is one of the state-of-the-art methods for cross-lingual

[1] https://github.com/shimo-lab/kadingir

word embeddings based on the skip-gram model. We obtain vector representations of words using publicly available implementation.[2]

4.3 Results

In CL-Eigenwords, vocabulary size $V^{(1)} = V^{(2)} = 10^4$, window size $h^{(1)} = h^{(2)} = 2$, the constant $b^{(1)} = b^{(2)} = 10^3$. The dimensionality of vector representations is $K = 40, 100$, or 200. Similarities of two vector representations are measured by the unweighted cosine similarity in CL-LSI and BilBOWA. Our experiments were performed on a CentOS 7.2 server with Intel Xeon E5-2680 v3 CPU, 256GB of RAM and gcc 4.8.5. The computation times and the result accuracies of word translation tasks are shown in Table 1. We observe that CL-Eigenwords is competitive with BilBOWA and CL-LSI. In particular, CL-Eigenwords performed very well for the most frequent words (ranks 1–1000) in this particular parameter setting. Furthermore, the computation times of CL-Eigenwords are as short as those of BilBOWA for achieving similar accuracies. Preliminary experiments also suggest that CL-Eigenwords works well for semi-supervised learning where sentence-alignment is specified only partially; the word translation accuracies are maintained well with aligned 240K lines and unaligned 260K lines.

5 Conclusion

We proposed CL-Eigenwords for incorporating cross-lingual information into the monolingual Eigenwords. Although our method is simple, experimental results of English-Spanish word translation tasks show that the proposed method is competitive with other state-of-the-art cross-lingual methods.

Acknowledgments

This work was partially supported by grants from Japan Society for the Promotion of Science KAKENHI (24300106, 16H01547 and 16H02789) to HS.

Appendix

In this Appendix, we discuss the relationships between CL-LSI and CL-Eigenwords.

[2] https://github.com/gouwsmeister/bilbowa

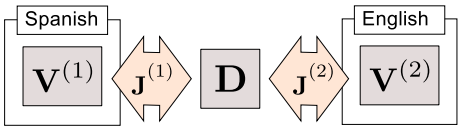

Figure 3: Cross-Language Latent Semantic Indexing (CL-LSI) does not use the context information.

Let $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{D}, \mathbf{J}^{(1)}, \mathbf{J}^{(2)}$ be those defined in Section 3. In CL-LSI, we consider the truncated singular value decomposition of a word-document matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{V}^{(1)\top}\mathbf{J}^{(1)} \\ \mathbf{V}^{(2)\top}\mathbf{J}^{(2)} \end{pmatrix} \approx \mathbf{A}_V \mathbf{\Lambda}_K \mathbf{A}_D^\top$$

using the largest K singular values. Then row vectors of $\mathbf{A}_V$ are the vector representations of words of CL-LSI.

CL-LSI can also be interpreted as an eigenvalue decomposition of $\mathbf{H} = \mathbf{X}^\top \mathbf{W} \mathbf{X}$ where

$$\mathbf{X} = \begin{pmatrix} \mathbf{V}^{(1)} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{V}^{(2)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{D} \end{pmatrix},$$
$$\mathbf{W} = \begin{pmatrix} \mathbf{O} & \mathbf{O} & \mathbf{J}^{(1)} \\ \mathbf{O} & \mathbf{O} & \mathbf{J}^{(2)} \\ \mathbf{J}^{(1)\top} & \mathbf{J}^{(2)\top} & \mathbf{O} \end{pmatrix}$$

are redefined from those in Section 3 by removing submatrices related to contexts. The structure of $\mathbf{X}$ and $\mathbf{W}$ is illustrated in Fig. 3. Similarly to CL-Eigenwords of Section 3, but ignoring $\mathbf{G}$, we define $\mathbf{A} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)$ with the eigenvectors of $\mathbf{H}$ for the largest K eigenvalues $\lambda_1, \ldots, \lambda_K$. It then follows from

$$\mathbf{H} = \begin{pmatrix} \mathbf{O} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{O} \end{pmatrix}$$

that $\mathbf{A}^\top = 2^{-1/2}(\mathbf{A}_V^\top, \mathbf{A}_D^\top)$ with the same $\mathbf{A}_V$ and $\mathbf{A}_D$ obtained by the truncated singular value decomposition. The eigenvalues are the same as the singular values: $\mathrm{diag}(\lambda_1, \ldots, \lambda_K) = \mathbf{\Lambda}_K$. Therefore CL-LSI is interpreted as a variant of CL-Eigenwords without the context information.

References

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal, September. Association for Computational Linguistics.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster, and Lyle H. Ungar. 2012. Two step cca: A new spectral method for estimating vector models of words. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 1551–1558, New York, NY, USA, July. Omnipress.

Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 748–756.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.

Angeliki Lazaridou, The Nghia Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. 1998. Automatic cross-language information retrieval using latent semantic indexing. In *Cross-language information retrieval*, pages 51–62. Springer.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nozomi Nori, Danushka Bollegala, and Hisashi Kashima. 2012. Multinomial relation prediction in social data: A dimension reduction approach. In *AAAI*, volume 12, pages 115–121.

Tianze Shi, Zhiyuan Liu, Yang Liu, and Maosong Sun. 2015. Learning cross-lingual word embeddings via matrix co-factorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 567–572, Beijing, China, July. Association for Computational Linguistics.

Hidetoshi Shimodaira. 2016. Cross-validation of matching correlation analysis by resampling matching weights. *Neural Networks*, 75:126–140.

Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, Jan.