Model-based Word Embeddings from Decompositions of Count Matrices

Karl Stratos **Michael Collins** **Daniel Hsu**

Columbia University, New York, NY 10027, USA
{stratos, mcollins, djhsu}@cs.columbia.edu

Abstract

This work develops a new statistical understanding of word embeddings induced from transformed count data. Using the class of hidden Markov models (HMMs) underlying Brown clustering as a generative model, we demonstrate how canonical correlation analysis (CCA) and certain count transformations permit efficient and effective recovery of model parameters with lexical semantics. We further show in experiments that these techniques empirically outperform existing spectral methods on word similarity and analogy tasks, and are also competitive with other popular methods such as WORD2VEC and GLOVE.

1 Introduction

The recent spike of interest in dense, low-dimensional lexical representations—i.e., word embeddings—is largely due to their ability to capture subtle syntactic and semantic patterns that are useful in a variety of natural language tasks. A successful method for deriving such embeddings is the negative sampling training of the skip-gram model suggested by Mikolov et al. (2013b) and implemented in the popular software WORD2VEC. The form of its training objective was motivated by efficiency considerations, but has subsequently been interpreted by Levy and Goldberg (2014b) as seeking a *low-rank factorization* of a matrix whose entries are word-context co-occurrence counts, scaled and transformed in a certain way. This observation sheds new light on WORD2VEC, yet also raises several new questions about word embeddings based on decomposing count data. What is the right matrix to decompose? Are there rigorous justifications for the choice of matrix and count transformations?

In this paper, we answer some of these questions by investigating the decomposition specified by CCA (Hotelling, 1936), a powerful technique for inducing generic representations whose computation is efficiently and exactly reduced to that of a matrix singular value decomposition (SVD). We build on and strengthen the work of Stratos et al. (2014) which uses CCA for learning the class of HMMs underlying Brown clustering. We show that certain count transformations enhance the accuracy of the estimation method and significantly improve the empirical performance of word representations derived from these model parameters (Table 1).

In addition to providing a rigorous justification for CCA-based word embeddings, we also supply a general template that encompasses a range of spectral methods (algorithms employing SVD) for inducing word embeddings in the literature, including the method of Levy and Goldberg (2014b). In experiments, we demonstrate that CCA combined with the square-root transformation achieves the best result among spectral methods and performs competitively with other popular methods such as WORD2VEC and GLOVE on word similarity and analogy tasks. We additionally demonstrate that CCA embeddings provide the most competitive improvement when used as features in named-entity recognition (NER).

2 Notation

We use $[n]$ to denote the set of integers $\{1, \ldots, n\}$. We denote the $m \times m$ diagonal matrix with values $v_1 \ldots v_m$ along the diagonal by $\text{diag}(v_1 \ldots v_m)$. We write $[a_1 \ldots a_m]$ to denote a matrix whose i-th column is a_i. The expected value of a random variable X is denoted by $\mathbf{E}[X]$. Given a matrix Ω and an exponent a, we distinguish the entrywise power operation $\Omega^{\langle a \rangle}$ (i.e., $\Omega^{\langle a \rangle}_{i,j} = (\Omega_{i,j})^a$) from the matrix power operation Ω^a (defined only for square Ω).

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1282–1291, Beijing, China, July 26-31, 2015.

3 Background in CCA

In this section, we review the variational characterization of CCA. This provides a flexible framework for a wide variety of tasks. CCA seeks to maximize a statistical quantity known as the Pearson correlation coefficient between random variables $L, R \in \mathbb{R}$:

$$\text{Cor}(L, R) := \frac{\mathbf{E}[LR] - \mathbf{E}[L]\mathbf{E}[R]}{\sqrt{\mathbf{E}[L^2] - \mathbf{E}[L]^2}\sqrt{\mathbf{E}[R^2] - \mathbf{E}[R]^2}}$$

This is a value in $[-1, 1]$ indicating the degree of linear dependence between L and R.

3.1 CCA objective

Let $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^{n'}$ be two random vectors. Without loss of generality, we will assume that X and Y have zero mean.[1] Let $m \leq \min(n, n')$. CCA can be cast as finding a set of projection vectors (called canonical directions) $a_1 \ldots a_m \in \mathbb{R}^n$ and $b_1 \ldots b_m \in \mathbb{R}^{n'}$ such that for $i = 1 \ldots m$:

$$\begin{aligned}(a_i, b_i) = \underset{a \in \mathbb{R}^n,\, b \in \mathbb{R}^{n'}}{\arg\max} \quad & \text{Cor}(a^\top X, b^\top Y) \qquad (1)\\ \text{Cor}(a^\top X, a_j^\top X) = 0 \quad & \forall j < i \\ \text{Cor}(b^\top Y, b_j^\top Y) = 0 \quad & \forall j < i\end{aligned}$$

That is, at each i we simultaneously optimize vectors a, b so that the projected random variables $a^\top X, b^\top Y \in \mathbb{R}$ are maximally correlated, subject to the constraint that the projections are uncorrelated to all previous projections.

Let $A := [a_1 \ldots a_m]$ and $B := [b_1 \ldots b_m]$. Then we can think of the joint projections

$$\underline{X} = A^\top X \qquad \underline{Y} = B^\top Y \qquad (2)$$

as new m-dimensional representations of the original variables that are transformed to be as correlated as possible with each other. Furthermore, often $m \ll \min(n, n')$, leading to a dramatic reduction in dimensionality.

3.2 Exact solution via SVD

Eq. (1) is non-convex due to the terms a and b that interact with each other, so it cannot be solved exactly using a standard optimization technique. However, a method based on SVD provides an efficient and exact solution. See Hardoon et al. (2004) for a detailed discussion.

[1] This can be always achieved through data preprocessing ("centering").

Lemma 3.1 (Hotelling (1936))**.** *Assume X and Y have zero mean. The solution (A, B) to (1) is given by $A = \boldsymbol{E}[XX^\top]^{-1/2}U$ and $B = \boldsymbol{E}[YY^\top]^{-1/2}V$ where the i-th column of $U \in \mathbb{R}^{n \times m}$ ($V \in \mathbb{R}^{n' \times m}$) is the left (right) singular vector of*

$$\Omega := \boldsymbol{E}[XX^\top]^{-1/2}\boldsymbol{E}[XY^\top]\boldsymbol{E}[YY^\top]^{-1/2} \qquad (3)$$

corresponding to the i-th largest singular value σ_i. Furthermore, $\sigma_i = Cor(a_i^\top X, b_i^\top Y)$.

3.3 Using CCA for word representations

As presented in Section 3.1, CCA is a general framework that operates on a pair of random variables. Adapting CCA specifically to inducing word representations results in a simple recipe for calculating (3).

A natural approach is to set X to represent a word and Y to represent the relevant "context" information about a word. We can use CCA to project X and Y to a low-dimensional space in which they are maximally correlated: see Eq. (2). The projected X can be considered as a new word representation.

Denote the set of distinct word types by $[n]$. We set $X, Y \in \mathbb{R}^n$ to be one-hot encodings of words and their associated context words. We define a context word to be a word occurring within ρ positions to the left and right (excluding the current word). For example, with $\rho = 1$, the following snippet of text where the current word is "souls":

```
Whatever our souls are made of
```

will generate two samples of $X \times Y$: a pair of indicator vectors for "souls" and "our", and a pair of indicator vectors for "souls" and "are".

CCA requires performing SVD on the following matrix $\Omega \in \mathbb{R}^{n \times n}$:

$$\begin{aligned}\Omega = &(\mathbf{E}[XX^\top] - \mathbf{E}[X]\mathbf{E}[X]^\top)^{-1/2} \\ &(\mathbf{E}[XY^\top] - \mathbf{E}[X]\mathbf{E}[Y]^\top) \\ &(\mathbf{E}[YY^\top] - \mathbf{E}[Y]\mathbf{E}[Y]^\top)^{-1/2}\end{aligned}$$

At a quick glance, this expression looks daunting: we need to perform matrix inversion and multiplication on potentially large dense matrices. However, Ω is easily computable with the following observations:

Observation 1. We can ignore the centering operation when the sample size is large (Dhillon et al.,

2011). To see why, let $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ be N samples of X and Y. Consider the sample estimate of the term $\mathbf{E}[XY^\top] - \mathbf{E}[X]\mathbf{E}[Y]^\top$:

$$\frac{1}{N}\sum_{i=1}^{N} x^{(i)}(y^{(i)})^\top - \frac{1}{N^2}\left(\sum_{i=1}^{N} x^{(i)}\right)\left(\sum_{i=1}^{N} y^{(i)}\right)^\top$$

The first term dominates the expression when N is large. This is indeed the setting in this task where the number of samples (word-context pairs in a corpus) easily tends to billions.

Observation 2. The (uncentered) covariance matrices $\mathbf{E}[XX^\top]$ and $\mathbf{E}[YY^\top]$ are diagonal. This follows from our definition of the word and context variables as one-hot encodings since $\mathbf{E}[X_w X_{w'}] = 0$ for $w \neq w'$ and $\mathbf{E}[Y_c Y_{c'}] = 0$ for $c \neq c'$.

With these observations and the binary definition of (X, Y), each entry in Ω now has a simple closed-form solution:

$$\Omega_{w,c} = \frac{P(X_w = 1, Y_c = 1)}{\sqrt{P(X_w = 1)P(Y_c = 1)}} \qquad (4)$$

which can be readily estimated from a corpus.

4 Using CCA for parameter estimation

In a less well-known interpretation of Eq. (4), CCA is seen as a parameter estimation algorithm for a language model (Stratos et al., 2014). This model is a restricted class of HMMs introduced by Brown et al. (1992), henceforth called the Brown model. In this section, we extend the result of Stratos et al. (2014) and show that its correctness is preserved under certain element-wise data transformations.

4.1 Clustering under a Brown model

A Brown model is a 5-tuple (n, m, π, t, o) for $n, m \in \mathbb{N}$ and functions π, t, o where

- $[n]$ is a set of word types.
- $[m]$ is a set of hidden states.
- $\pi(h)$ is the probability of generating $h \in [m]$ in the first position of a sequence.
- $t(h'|h)$ is the probability of generating $h' \in [m]$ given $h \in [m]$.
- $o(w|h)$ is the probability of generating $w \in [n]$ given $h \in [m]$.

Importantly, the model makes the following additional assumption:

Assumption 4.1 (Brown assumption)**.** *For each word type $w \in [n]$, there is a unique hidden state $\mathcal{H}(w) \in [m]$ such that $o(w|\mathcal{H}(w)) > 0$ and $o(w|h) = 0$ for all $h \neq \mathcal{H}(w)$.*

In other words, this model is an HMM in which observation states are partitioned by hidden states. Thus a sequence of N words $w_1 \dots w_N \in [n]^N$ has probability $\pi(\mathcal{H}(w_1)) \times \prod_{i=1}^N o(w_i|\mathcal{H}(w_i)) \times \prod_{i=1}^{N-1} t(\mathcal{H}(w_{i+1})|\mathcal{H}(w_i))$.

An equivalent definition of a Brown model is given by organizing the parameters in matrix form. Under this definition, a Brown model has parameters (π, T, O) where $\pi \in \mathbb{R}^m$ is a vector and $T \in \mathbb{R}^{m \times m}, O \in \mathbb{R}^{n \times m}$ are matrices whose entries are set to:

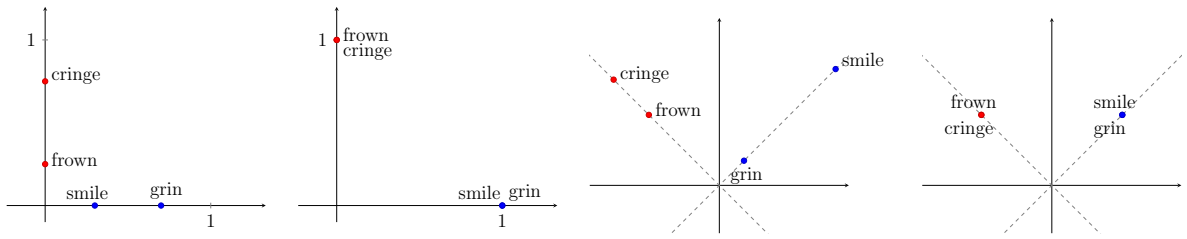
$$\begin{aligned} \pi_h &= \pi(h) && h \in [m] \\ T_{h',h} &= t(h'|h) && h, h' \in [m] \\ O_{w,h} &= o(w|h) && h \in [m], w \in [n] \end{aligned}$$

Our main interest is in obtaining some representations of word types that allow us to identify their associated hidden states under the model. For this purpose, representing a word by the corresponding row of O is sufficient. To see this, note that each row of O must have a single nonzero entry by Assumption 4.1. Let $v(w) \in \mathbb{R}^m$ be the w-th row of O normalized to have unit 2-norm: then $v(w) = v(w')$ iff $\mathcal{H}(w) = \mathcal{H}(w')$. See Figure 1(a) for illustration.

A crucial aspect of this representational scheme is that its correctness is *invariant* to scaling and rotation. In particular, clustering the normalized rows of $\text{diag}(s)O^{\langle a \rangle}\text{diag}(s_2)Q^\top$ where $O^{\langle a \rangle}$ is any element-wise power of O with any $a \neq 0$, $Q \in \mathbb{R}^{m \times m}$ is any orthogonal transformation, and $s_1 \in \mathbb{R}^n$ and $s_2 \in \mathbb{R}^m$ are any positive vectors yields the correct clusters under the model. See Figure 1(b) for illustration.

4.2 Spectral estimation

Thus we would like to estimate O and use its rows for representing word types. But the likelihood function under the Brown model is non-convex, making an MLE estimation of the model parameters difficult. However, the hard-clustering assumption (Assumption 4.1) allows for a simple



(a) (b)

Figure 1: Visualization of the representational scheme under a Brown model with 2 hidden states. (a) Normalizing the original rows of O. (b) Normalizing the scaled and rotated rows of O.

spectral method for consistent parameter estimation of O.

To state the theorem, we define an additional quantity. Let ρ be the number of left/right context words to consider in CCA. Let $(H_1, \ldots, H_N) \in [m]^N$ be a random sequence of hidden states drawn from the Brown model where $N \geq 2\rho + 1$. Independently, pick a position $I \in [\rho + 1, N - \rho]$ uniformly at random. Define $\tilde{\pi} \in \mathbb{R}^m$ where $\tilde{\pi}_h := P(H_I = h)$ for each $h \in [m]$.

Theorem 4.1. *Assume $\tilde{\pi} > 0$ and rank(O) = rank(T) = m. Assume that a Brown model (π, T, O) generates a sequence of words. Let $X, Y \in \mathbb{R}^n$ be one-hot encodings of words and their associated context words. Let $U \in \mathbb{R}^{n \times m}$ be the matrix of m left singular vectors of $\Omega^{\langle a \rangle} \in \mathbb{R}^{n \times n}$ corresponding to nonzero singular values where Ω is defined in Eq. (4) and $a \neq 0$:*

$$\Omega^{\langle a \rangle}_{w,c} = \frac{P(X_w = 1, Y_c = 1)^a}{\sqrt{P(X_w = 1)^a P(Y_c = 1)^a}}$$

Then there exists an orthogonal matrix $Q \in \mathbb{R}^{m \times m}$ and a positive $s \in \mathbb{R}^m$ such that $U = O^{\langle a/2 \rangle} \mathrm{diag}(s) Q^\top$.

This theorem states that the CCA projection of words in Section 3.3 is the rows of O up to scaling and rotation even if we raise each element of Ω in Eq. (4) to an arbitrary (nonzero) power. The proof is a variant of the proof in Stratos et al. (2014) and is given in Appendix A.

4.3 Choice of data transformation

Given a corpus, the sample estimate of $\Omega^{\langle a \rangle}$ is given by:

$$\hat{\Omega}^{\langle a \rangle}_{w,c} = \frac{\#(w,c)^a}{\sqrt{\#(w)^a \#(c)^a}} \tag{5}$$

where $\#(w, c)$ denotes the co-occurrence count of word w and context c in the corpus, $\#(w) := \sum_c \#(w, c)$, and $\#(c) := \sum_w \#(w, c)$. What choice of a is beneficial and why? We use $a = 1/2$ for the following reason: it stabilizes the variance of the term and thereby gives a more statistically stable solution.

4.3.1 Variance stabilization for word counts

The square-root transformation is a *variance-stabilizing transformation* for Poisson random variables (Bartlett, 1936; Anscombe, 1948). In particular, the square-root of a Poisson variable has variance close to $1/4$, independent of its mean.

Lemma 4.1 (Bartlett (1936))**.** *Let X be a random variable with distribution Poisson$(n \times p)$ for any $p \in (0, 1)$ and positive integer n. Define $Y := \sqrt{X}$. Then the variance of Y approaches $1/4$ as $n \to \infty$.*

This transformation is relevant for word counts because they can be naturally modeled as Poisson variables. Indeed, if word counts in a corpus of length N are drawn from a multinomial distribution over $[n]$ with N observations, then these counts have the same distribution as n independent Poisson variables (whose rate parameters are related to the multinomial probabilities), conditioned on their sum equaling N (Steel, 1953). Empirically, the peaky concentration of a Poisson distribution is well-suited for modeling word occurrences.

4.3.2 Variance-weighted squared-error minimization

At the heart of CCA is computing the SVD of the $\Omega^{\langle a \rangle}$ matrix: this can be interpreted as solving the following (non-convex) squared-error minimization problem:

$$\min_{u_w, v_c \in \mathbb{R}^m} \sum_{w,c} \left(\Omega^{\langle a \rangle}_{w,c} - u_w^\top v_c \right)^2$$

But we note that minimizing *unweighted* squared-error objectives is generally suboptimal when the target values are heteroscedastic. For instance, in linear regression, it is well-known that a *weighted least squares* estimator dominates ordinary least squares in terms of statistical efficiency (Aitken, 1936; Lehmann and Casella, 1998). For our setting, the analogous weighted least squares optimization is:

$$\min_{u_w, v_c \in \mathbb{R}^m} \sum_{w,c} \frac{1}{\text{Var}\left(\Omega^{\langle a \rangle}_{w,c}\right)} \left(\Omega^{\langle a \rangle}_{w,c} - u_w^\top v_c\right)^2 \quad (6)$$

where $\text{Var}(X) := \mathbf{E}[X^2] - \mathbf{E}[X]^2$. This optimization is, unfortunately, generally intractable (Srebro et al., 2003). The square-root transformation, nevertheless, obviates the variance-based weighting since the target values have approximately the same variance of 1/4.

5 A template for spectral methods

Figure 2 gives a generic template that encompasses a range of spectral methods for deriving word embeddings. All of them operate on co-occurrence counts $\#(w, c)$ and share the low-rank SVD step, but they can differ in the data transformation method (t) and the definition of the matrix of scaled counts for SVD (s).

We introduce two additional parameters $\alpha, \beta \leq 1$ to account for the following details. Mikolov et al. (2013b) proposed smoothing the empirical context distribution as $\hat{p}_\alpha(c) := \#(c)^\alpha / \sum_c \#(c)^\alpha$ and found $\alpha = 0.75$ to work well in practice. We also found that setting $\alpha = 0.75$ gave a small but consistent improvement over setting $\alpha = 1$. Note that the choice of α only affects methods that make use of the context distribution ($s \in \{\text{ppmi}, \text{cca}\}$).

The parameter β controls the role of singular values in word embeddings. This is always 0 for CCA as it does not require singular values. But for other methods, one can consider setting $\beta > 0$ since the best-fit subspace for the rows of Ω is given by $U\Sigma$. For example, Deerwester et al. (1990) use $\beta = 1$ and Levy and Goldberg (2014b) use $\beta = 0.5$. However, it has been found by many (including ourselves) that setting $\beta = 1$ yields substantially worse representations than setting $\beta \in \{0, 0.5\}$ (Levy et al., 2015).

Different combinations of these aspects reproduce various spectral embeddings explored in the literature. We enumerate some meaningful combinations:

SPECTRAL-TEMPLATE
Input: word-context co-occurrence counts $\#(w, c)$, dimension m, transformation method t, scaling method s, context smoothing exponent $\alpha \leq 1$, singular value exponent $\beta \leq 1$
Output: vector $v(w) \in \mathbb{R}^m$ for each word $w \in [n]$
Definitions: $\#(w) := \sum_c \#(w, c)$, $\#(c) := \sum_w \#(w, c)$, $N(\alpha) := \sum_c \#(c)^\alpha$

1. Transform all $\#(w, c)$, $\#(w)$, and $\#(c)$:

$$\#(\cdot) \leftarrow \begin{cases} \#(\cdot) & \text{if } t = \text{—} \\ \log(1 + \#(\cdot)) & \text{if } t = \text{log} \\ \#(\cdot)^{2/3} & \text{if } t = \text{two-thirds} \\ \sqrt{\#(\cdot)} & \text{if } t = \text{sqrt} \end{cases}$$

2. Scale statistics to construct a matrix $\Omega \in \mathbb{R}^{n \times n}$:

$$\Omega_{w,c} \leftarrow \begin{cases} \#(w, c) & \text{if } s = \text{—} \\ \frac{\#(w,c)}{\#(w)} & \text{if } s = \text{reg} \\ \max\left(\log \frac{\#(w,c)N(\alpha)}{\#(w)\#(c)^\alpha}, 0\right) & \text{if } s = \text{ppmi} \\ \frac{\#(w,c)}{\sqrt{\#(w)\#(c)^\alpha}} \sqrt{\frac{N(\alpha)}{N(1)}} & \text{if } s = \text{cca} \end{cases}$$

3. Perform rank-m SVD on $\Omega \approx U\Sigma V^\top$ where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_m)$ is a diagonal matrix of ordered singular values $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$.

4. Define $v(w) \in \mathbb{R}^m$ to be the w-th row of $U\Sigma^\beta$ normalized to have unit 2-norm.

Figure 2: A template for spectral word embedding methods.

No scaling $\left[t \in \{\text{—}, \text{log}, \text{sqrt}\},\ s = \text{—}\right]$. This is a commonly considered setting (e.g., in Pennington et al. (2014)) where no scaling is applied to the co-occurrence counts. It is however typically accompanied with some kind of data transformation.

Positive point-wise mutual information (PPMI) $\left[t = \text{—},\ s = \text{ppmi}\right]$. Mutual information is a popular metric in many natural language tasks (Brown et al., 1992; Pantel and Lin, 2002). In this setting, each term in the matrix for SVD is set as the point-wise mutual information between word w and context c:

$$\log \frac{\hat{p}(w, c)}{\hat{p}(w)\hat{p}_\alpha(c)} = \log \frac{\#(w, c) \sum_c \#(c)^\alpha}{\#(w)\#(c)^\alpha}$$

Typically negative values are thresholded to 0 to keep Ω sparse. Levy and Goldberg (2014b) observed that the negative sampling objective of the skip-gram model of Mikolov et al. (2013b) is implicitly factorizing a shifted version of this matrix.[2]

[2]This is not equivalent to applying SVD on this matrix, however, since the loss function is different.

Regression $\left[t \in \{—, \text{sqrt}\},\ s = \text{reg}\right]$. Another novelty of our work is considering a low-rank approximation of a linear regressor that predicts the context from words. Denoting the word sample matrix by $\mathcal{X} \in \mathbb{R}^{N \times n}$ and the context sample matrix by $\mathcal{Y} \in \mathbb{R}^{N \times n}$, we seek $U^* = \arg\min_{U \in \mathbb{R}^{n \times n}} ||\mathcal{Y} - \mathcal{X}U||^2$ whose closed-from solution is given by:

$$U^* = (\mathcal{X}^\top \mathcal{X})^{-1} \mathcal{X}^\top \mathcal{Y} \tag{7}$$

Thus we aim to compute a low-rank approximation of U^* with SVD. This is inspired by other predictive models in the representation learning literature (Ando and Zhang, 2005; Mikolov et al., 2013a). We consider applying the square-root transformation for the same variance stabilizing effect discussed in Section 4.3.

CCA $\left[t \in \{—, \text{two-thirds}, \text{sqrt}\},\ s = \text{cca}\right]$. This is the focus of our work. As shown in Theorem 4.1, we can take the element-wise power transformation on counts (such as the power of $1, 2/3, 1/2$ in this template) while preserving the representational meaning of word embeddings under the Brown model interpretation. If there is no data transformation ($t = —$), then we recover the original spectral algorithm of Stratos et al. (2014).

6 Related work

We make a few remarks on related works not already discussed earlier. Dhillon et al. (2011) and (2012) propose novel modifications of CCA (LR-MVL and two-step CCA) to derive word embeddings, but do not establish any explicit connection to learning HMM parameters or justify the square-root transformation. Pennington et al. (2014) propose a weighted factorization of log-transformed co-occurrence counts, which is generally an intractable problem (Srebro et al., 2003). In contrast, our method requires only efficiently computable matrix decompositions. Finally, word embeddings have also been used as features to improve performance in a variety of supervised tasks such as sequence labeling (Dhillon et al., 2011; Collobert et al., 2011) and dependency parsing (Lei et al., 2014; Chen and Manning, 2014). Here, we focus on understanding word embeddings in the context of a generative word class model, as well as in empirical tasks that directly evaluate the word embeddings themselves.

7 Experiments

7.1 Word similarity and analogy

We first consider word similarity and analogy tasks for evaluating the quality of word embeddings. Word similarity measures the Spearman's correlation coefficient between the human scores and the embeddings' cosine similarities for word pairs. Word analogy measures the accuracy on syntactic and semantic analogy questions. We refer to Levy and Goldberg (2014a) for a detailed description of these tasks. We use the multiplicative technique of Levy and Goldberg (2014a) for answering analogy questions.

For the choice of corpus, we use a preprocessed English Wikipedia dump (`http://dumps.wikimedia.org/`). The corpus contains around 1.4 billion words. We only preserve word types that appear more than 100 times and replace all others with a special symbol, resulting in a vocabulary of size around 188k. We define context words to be 5 words to the left/right for all considered methods.

We use three word similarity datasets each containing 353, 3000, and 2034 word pairs.[3] We report the average similarity score across these datasets under the label AVG-SIM. We use two word analogy datasets that we call SYN (8000 syntactic analogy questions) and MIXED (19544 syntactic and semantic analogy questions).[4]

We implemented the template in Figure 2 in C++.[5] We compared against the public implementation of WORD2VEC by Mikolov et al. (2013b) and GLOVE by Pennington et al. (2014). These external implementations have numerous hyperparameters that are not part of the core algorithm, such as random subsampling in WORD2VEC and the word-context averaging in GLOVE. We refer to Levy et al. (2015) for a discussion of the effect of these features. In our experiments, we enable all these features with the recommended default settings.

We reserve a half of each dataset (by category)

[3] WordSim-353: `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`; MEN: `http://clic.cimec.unitn.it/~elia.bruni/MEN.html`; Stanford Rare Word: `http://www-nlp.stanford.edu/~lmthang/morphoNLM/`.

[4] `http://research.microsoft.com/en-us/um/people/gzweig/Pubs/myz_naacl13_test_set.tgz`; `http://www.fit.vutbr.cz/~imikolov/rnnlm/word-test.v1.txt`

[5] The code is available at `https://github.com/karlstratos/singular`.

Configuration		500 dimensions			1000 dimensions		
Transform (t)	Scale (s)	AVG-SIM	SYN	MIXED	AVG-SIM	SYN	MIXED
—	—	0.514	31.58	28.39	0.522	29.84	32.15
sqrt	—	0.656	60.77	65.84	0.646	57.46	64.97
log	—	0.669	59.28	66.86	0.672	55.66	68.62
—	reg	0.530	29.61	36.90	0.562	32.78	37.65
sqrt	reg	0.625	63.97	67.30	0.638	**65.98**	70.04
—	ppmi	0.638	41.62	58.80	0.665	47.11	65.34
sqrt	cca	**0.678**	**66.40**	**74.73**	**0.690**	65.14	**77.70**

Table 2: Performance of various spectral methods on the development portion of data.

Transform (t)	AVG-SIM	SYN	MIXED
—	0.572	39.68	57.64
log	0.675	55.61	69.26
two-thirds	0.650	60.52	74.00
sqrt	**0.690**	**65.14**	**77.70**

Table 1: Performance of CCA (1000 dimensions) on the development portion of data with different data transformation methods ($\alpha = 0.75$, $\beta = 0$).

as a held-out portion for development and use the other half for final evaluation.

7.1.1 Effect of data transformation for CCA

We first look at the effect of different data transformations on the performance of CCA. Table 1 shows the result on the development portion with 1000-dimensional embeddings. We see that without any transformation, the performance can be quite bad—especially in word analogy. But there is a marked improvement upon transforming the data. Moreover, the square-root transformation gives the best result, improving the accuracy on the two analogy datasets by 25.46% and 20.06% in absolute magnitude. This aligns with the discussion in Section 4.3.

7.1.2 Comparison among different spectral embeddings

Next, we look at the performance of various combinations in the template in Figure 2. We smooth the context distribution with $\alpha = 0.75$ for PPMI and CCA. We use $\beta = 0.5$ for PPMI (which has a minor improvement over $\beta = 0$) and $\beta = 0$ for all other methods. We generally find that using $\beta = 0$ is critical to obtaining good performance for $s \in \{—, \text{reg}\}$.

Table 2 shows the result on the development portion for both 500 and 1000 dimensions. Even without any scaling, SVD performs reasonably well with the square-root and log transformations. The regression scaling performs very poorly without data transformation, but once the square-root transformation is applied it performs quite well (especially in analogy questions). The PPMI scaling achieves good performance in word similarity but not in word analogy. The CCA scaling, combined with the square-root transformation, gives the best overall performance. In particular, it performs better than all other methods in mixed analogy questions by a significant margin.

7.1.3 Comparison with other embedding methods

We compare spectral embedding methods against WORD2VEC and GLOVE on the test portion. We use the following combinations based on their performance on the development portion:

- LOG: log transform, — scaling
- REG: sqrt transform, reg scaling
- PPMI: — transform, ppmi scaling
- CCA: sqrt transform, cca scaling

For WORD2VEC, there are two model options: continuous bag-of-words (CBOW) and skip-gram (SKIP). Table 3 shows the result for both 500 and 1000 dimensions.

In word similarity, spectral methods generally excel, with CCA consistently performing the best. SKIP is the only external package that performs comparably, with GLOVE and CBOW falling behind. In word analogy, REG and CCA are significantly better than other spectral methods. They are also competitive to GLOVE and CBOW, but SKIP does perform the best among all compared methods on (especially syntactic) analogy questions.

Method		500 dimensions			1000 dimensions		
		AVG-SIM	SYN	MIXED	AVG-SIM	SYN	MIXED
Spectral	LOG	0.652	59.52	67.27	0.635	56.53	68.67
	REG	0.602	65.51	67.88	0.609	66.47	70.48
	PPMI	0.628	43.81	58.38	0.637	48.99	63.82
	CCA	**0.655**	68.38	74.17	**0.650**	66.08	76.38
Others	GLOVE	0.576	68.30	78.08	0.586	67.40	78.73
	CBOW	0.597	75.79	73.60	0.509	70.97	60.12
	SKIP	0.642	**81.08**	**78.73**	0.641	**79.98**	**83.35**

Table 3: Performance of different word embedding methods on the test portion of data. See the main text for the configuration details of spectral methods.

7.2 As features in a supervised task

Finally, we use word embeddings as features in NER and compare the subsequent improvements between various embedding methods. The experimental setting is identical to that of Stratos et al. (2014). We use the Reuters RCV1 corpus which contains 205 million words. With frequency thresholding, we end up with a vocabulary of size around 301k. We derive LOG, REG, PPMI, and CCA embeddings as described in Section 7.1.3, and GLOVE, CBOW, and SKIP embeddings again with the recommended default settings. The number of left/right contexts is 2 for all methods. For comparison, we also derived 1000 Brown clusters (BROWN) on the same vocabulary and used the resulting bit strings as features (Brown et al., 1992).

Table 4 shows the result for both 30 and 50 dimensions. In general, using any of these lexical features provides substantial improvements over the baseline.[6] In particular, the 30-dimensional CCA embeddings improve the F1 score by 2.84 on the development portion and by 4.88 on the test portion. All spectral methods perform competitively with external packages, with CCA and SKIP consistently delivering the biggest improvements on the development portion.

Features	30 dimensions		50 dimensions	
	Dev	Test	Dev	Test
—	90.04	84.40	90.04	84.40
BROWN	92.49	88.75	92.49	88.75
LOG	92.27	88.87	92.91	**89.67**
REG	92.51	88.08	92.73	88.88
PPMI	92.25	89.27	92.53	89.37
CCA	**92.88**	**89.28**	92.94	89.01
GLOVE	91.49	87.16	91.58	86.80
CBOW	92.44	88.34	92.83	89.21
SKIP	92.63	88.78	**93.11**	89.32

Table 4: NER F1 scores when word embeddings are added as features to the baseline (—).

8 Conclusion

In this work, we revisited SVD-based methods for inducing word embeddings. We examined a framework provided by CCA and showed that the resulting word embeddings can be viewed as cluster-revealing parameters of a certain model and that this result is robust to data transformation. Our proposed method gives the best result among spectral methods and is competitive to other popular word embedding techniques.

This work suggests many directions for future work. Past spectral methods that involved CCA without data transformation (e.g., Cohen et al. (2013)) may be revisited with the square-root transformation. Using CCA to induce representations other than word embeddings is another important future work. It would also be interesting to formally investigate the theoretical merits and algorithmic possibility of solving the variance-weighted objective in Eq. (6). Even though the objective is hard to optimize in the worst case, it may be tractable under natural conditions.

[6] We mention that the well-known dev/test discrepancy in the CoNLL 2003 dataset makes the results on the test portion less reliable.

Acknowledgments

We thank Omer Levy, Yoav Goldberg, and David Belanger for helpful discussions. This work was made possible by a research grant from Bloomberg's Knowledge Engineering team.

A Proof of Theorem 4.1

We first define some random variables. Let ρ be the number of left/right context words to consider in CCA. Let $(W_1, \ldots, W_N) \in [n]^N$ be a random sequence of words drawn from the Brown model where $N \geq 2\rho + 1$, along with the corresponding sequence of hidden states $(H_1, \ldots, H_N) \in [m]^N$. Independently, pick a position $I \in [\rho + 1, N - \rho]$ uniformly at random; pick an integer $J \in [-\rho, \rho] \backslash \{0\}$ uniformly at random. Define $B \in \mathbb{R}^{n \times n}$, $u, v \in \mathbb{R}^n$, $\tilde{\pi} \in \mathbb{R}^m$, and $\tilde{T} \in \mathbb{R}^{m \times m}$ as follows:

$$\begin{aligned}
B_{w,c} &:= P(W_I = w, W_{I+J} = c) && \forall w, c \in [n] \\
u_w &:= P(W_I = w) && \forall w \in [n] \\
v_c &:= P(W_{I+J} = c) && \forall c \in [n] \\
\tilde{\pi}_h &:= P(H_I = h) && \forall h \in [m] \\
\tilde{T}_{h',h} &:= P(H_{I+J} = h' | H_I = h) && \forall h, h' \in [m]
\end{aligned}$$

First, we show that $\Omega^{\langle a \rangle}$ has a particular structure under the Brown assumption. For the choice of positive vector $s \in \mathbb{R}^m$ in the theorem, we define $s_h := (\sum_w o(w|h)^a)^{-1/2}$ for all $h \in [m]$.

Lemma A.1. *$\Omega^{\langle a \rangle} = A\Theta^\top$ where $\Theta \in \mathbb{R}^{n \times m}$ has rank m and $A \in \mathbb{R}^{n \times m}$ is defined as:*

$$A := \mathit{diag}(O\tilde{\pi})^{-a/2} O^{\langle a \rangle} \mathit{diag}(\tilde{\pi})^{a/2} \mathit{diag}(s)$$

Proof. Let $\tilde{O} := O\tilde{T}$. It can be algebraically verified that $B = O\text{diag}(\tilde{\pi})\tilde{O}^\top$, $u = O\tilde{\pi}$, and $v = \tilde{O}\tilde{\pi}$. By Assumption 4.1, each entry of $B^{\langle a \rangle}$ has the form

$$\begin{aligned}
B^{\langle a \rangle}_{w,c} &= \left(\sum_{h \in [m]} O_{w,h} \times \tilde{\pi}_h \times \tilde{O}_{c,h} \right)^a \\
&= \left(O_{w,\mathcal{H}(w)} \times \tilde{\pi}_{\mathcal{H}(w)} \times \tilde{O}_{c,\mathcal{H}(w)} \right)^a \\
&= O^a_{w,\mathcal{H}(w)} \times \tilde{\pi}^a_{\mathcal{H}(w)} \times \tilde{O}^a_{c,\mathcal{H}(w)} \\
&= \sum_{h \in [m]} O^a_{w,h} \times \tilde{\pi}^a_h \times \tilde{O}^a_{c,h}
\end{aligned}$$

Thus $B^{\langle a \rangle} = O^{\langle a \rangle} \text{diag}(\tilde{\pi})^a (\tilde{O}^{\langle a \rangle})^\top$. Therefore,

$$\begin{aligned}
\Omega^{\langle a \rangle} &= \left(\text{diag}(u)^{-1/2} B \text{diag}(v)^{-1/2} \right)^{\langle a \rangle} \\
&= \text{diag}(u)^{-a/2} B^{\langle a \rangle} \text{diag}(v)^{-a/2} \\
&= \text{diag}(O\tilde{\pi})^{-a/2} O^{\langle a \rangle} \text{diag}(\tilde{\pi})^{a/2} \text{diag}(s) \\
&\quad \text{diag}(s)^{-1} \text{diag}(\tilde{\pi})^{a/2} (\tilde{O}^{\langle a \rangle})^\top \text{diag}(\tilde{O}\tilde{\pi})^{-a/2}
\end{aligned}$$

This gives the desired result. □

Next, we show that the left component of $\Omega^{\langle a \rangle}$ is in fact the emission matrix O up to (nonzero) scaling and is furthermore orthonormal.

Lemma A.2. *The matrix A in Lemma A.1 has the expression $A = O^{\langle a/2 \rangle} \mathit{diag}(s)$ and has orthonormal columns.*

Proof. By Assumption 4.1, each entry of A is simplified as follows:

$$\begin{aligned}
A_{w,h} &= \frac{o(w|h)^a \times \tilde{\pi}_h^{a/2} \times s_h}{o(w|\mathcal{H}(w))^{a/2} \times \tilde{\pi}^{a/2}_{\mathcal{H}(w)}} \\
&= o(w|h)^{a/2} \times s_h
\end{aligned}$$

This proves the first part of the lemma. Note that:

$$[A^\top A]_{h,h'} = \begin{cases} s_h^2 \times \sum_w o(w|h)^a & \text{if } h = h' \\ 0 & \text{otherwise} \end{cases}$$

Thus our choice of s gives $A^\top A = \mathcal{I}_{m \times m}$. □

Proof of Theorem 4.1. With Lemma A.1 and A.2, the proof is similar to the proof of Theorem 5.1 in Stratos et al. (2014). □

References

Alexander C Aitken. 1936. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853.

Francis J Anscombe. 1948. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, pages 246–254.

MSo Bartlett. 1936. The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society*, pages 68–78.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 740–750.

Shay B Cohen, Karl Stratos, Michael Collins, Dean P Foster, and Lyle H Ungar. 2013. Experiments with spectral learning of latent-variable pcfgs. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*, pages 148–157.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. 2011. Multi-view learning of word embeddings via cca. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 199–207.

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 *Proceedings of the International Conference on Machine learning*.

David Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664.

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

Erich Leo Lehmann and George Casella. 1998. *Theory of point estimation*, volume 31. Springer Science & Business Media.

Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. In *Proceedings of the Association for Computational Linguistics*, volume 1, pages 1381–1391.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Computational Natural Language Learning*, page 171.

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

Omer Levy, Yoav Goldberg, Ido Dagan, and Israel Ramat-Gan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3111–3119.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing*, volume 12.

Nathan Srebro, Tommi Jaakkola, et al. 2003. Weighted low-rank approximations. In *Proceedings of the International Conference on Machine learning*, volume 3, pages 720–727.

Robert G. D. Steel. 1953. Relation between poisson and multinomial distributions. Technical Report BU-39-M, Cornell University.

Karl Stratos, Do-kyum Kim, Michael Collins, and Daniel Hsu. 2014. A spectral algorithm for learning class-based n-gram models of natural language. In *Proceedings of the Association for Uncertainty in Artificial Intelligence*.