Multiview LSA: Representation Learning via Generalized CCA

Pushpendre Rastogi\textsuperscript{1} and Benjamin Van Durme\textsuperscript{1,2} and Raman Arora\textsuperscript{1}

\textsuperscript{1}Center for Language and Speech Processing
\textsuperscript{2}Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

Multiview LSA (MVLSA) is a generalization of Latent Semantic Analysis (LSA) that supports the fusion of arbitrary views of data and relies on Generalized Canonical Correlation Analysis (GCCA). We present an algorithm for fast approximate computation of GCCA, which when coupled with methods for handling missing values, is general enough to approximate some recent algorithms for inducing vector representations of words. Experiments across a comprehensive collection of test-sets show our approach to be competitive with the state of the art.

1 Introduction

Winograd (1972) wrote that: “Two sentences are paraphrases if they produce the same representation in the internal formalism for meaning”. This intuition is made soft in vector-space models (Turney and Pantel, 2010), where we say that expressions in language are paraphrases if their representations are close under some distance measure.

One of the earliest linguistic vector space models was Latent Semantic Analysis (LSA). LSA has been successfully used for Information Retrieval but it is limited in its reliance on a single matrix, or view, of term co-occurrences. Here we address the single-view limitation of LSA by demonstrating that the framework of Generalized Canonical Correlation Analysis (GCCA) can be used to perform Multiview LSA (MVLSA). This approach allows for the use of an arbitrary number of views in the induction process, including embeddings induced using other algorithms. We also present a fast approximate method for performing GCCA and approximately recover the objective of (Pennington et al., 2014) while accounting for missing values.

Our experiments show that MVLSA is competitive with state of the art approached for inducing vector representations of words and phrases. As a methodological aside, we discuss the (in-)significance of conclusions being drawn from comparisons done on small sized datasets.

2 Motivation

LSA is an application of Principal Component Analysis (PCA) to a term-document cooccurrence matrix. The principal directions found by PCA form the basis of the vector-space in which to represent the input terms (Landauer and Dumais, 1997). A drawback of PCA is that it can leverage only a single source of data and it is sensitive to scaling.

An arguably better approach to representation learning is Canonical Correlation Analysis (CCA) that induces representations that are maximally correlated across two views, allowing the utilization of two distinct sources of data. While an improvement over PCA, being limited to only two views is unfortunate in light of the fact that many sources of data (perspectives) are frequently available in practice. In such cases it is natural to extend CCA’s original objective of maximizing correlation between two views by maximizing some measure of the matrix $\Phi$ that contains all the pairwise correlations between linear projections of the covariates. This is how Generalized Canonical Correlation Analysis (GCCA) was first derived by Horst (1961). Recently these intuitive ideas about benefits of leveraging multiple sources of data have received strong theoretical backing due to the work by Sridharan and...
Kakade (2008) who showed that learning with multiple views is beneficial since it reduces the complexity of the learning problem by restricting the search space. Recent work by Anandkumar et al. (2014) showed that at least three views are necessary for recovering hidden variable models.

Note that there exist different variants of GCCA depending on the measure of $\Phi$ that we choose to maximize. Kettenring (1971) enumerated a variety of possible measures, such as the spectral-norm of $\Phi$. Kettenring noted that maximizing this spectral-norm is equivalent to finding linear projections of the covariates that are most amenable to rank-one PCA, or that can be best explained by a single term factor model. This variant was named MAX-VAR GCCA and was shown to be equivalent to a proposal by Carroll (1968), which searched for an auxiliary orthogonal representation $G$ that was maximally correlated to the linear projections of the covariates. Carroll’s objective targets the intuition that representations leveraging multiple views should correlate with all provided views as much as possible.

### 3 Proposed Method: MVLSA

Let $X_j \in \mathbb{R}^{N \times d_j}$ $\forall j \in [1, \ldots, J]$ be the mean centered matrix containing data from view $j$ such that row $i$ of $X_j$ contains the information for word $w_i$. Let the number of words in the vocabulary be $N$ and number of contexts (columns in $X_j$) be $d_j$. Following standard notation (Hastie et al., 2009) we call $X_j^\top X_j$ the scatter matrix and $X_j(X_j^\top X_j)^{-1}X_j^\top$ the projection matrix.

The objective of MAX-VAR GCCA can be written as the following optimization problem: Find $G \in \mathbb{R}^{N \times r}$ and $U_j \in \mathbb{R}^{d_j \times r}$ that solve:

$$\arg \min_{G, U_j} \sum_{j=1}^J \|G - X_j U_j\|_F^2,$$

subject to $G^\top G = I$.  

The matrix $G$ that solves problem (1) is our vector representation of the vocabulary. Finding $G$ reduces to spectral decomposition of sum of projection matrices of different views: Define

$$P_j = X_j (X_j^\top X_j)^{-1} X_j^\top,$$

$$M = \sum_{j=1}^J P_j.$$  

Then, for some positive diagonal matrix $\Lambda$, $G$ and $U_j$ satisfy:

$$MG = \Lambda,$$

$$U_j = (X_j^\top X_j)^{-1} X_j^\top G.$$  

Computationally storing $P_j \in \mathbb{R}^{N \times N}$ is problematic owing to memory constraints. Further, the scatter matrices may be non-singular leading to an ill-posed procedure. We now describe a novel scalable GCCA with $\ell_2$-regularization to address these issues.

**Approximate Regularized GCCA:** GCCA can be regularized by adding $r_j I$ to scatter matrix $X_j^\top X_j$ before doing the inversion where $r_j$ is a small constant e.g. $10^{-8}$. Projection matrices in (2) and (3) can then be written as

$$\tilde{P}_j = X_j (X_j^\top X_j + r_j I)^{-1} X_j^\top,$$

$$M = \sum_{j=1}^J \tilde{P}_j.$$  

Next, to scale up GCCA to large datasets, we first form a rank-$m$ approximation of projection matrices (Arora and Livescu, 2012) and then extend it to an eigendecomposition for $M$ following ideas by Savostyanov (2014). Consider the rank-$m$ SVD of $X_j$:

$$X_j = A_j S_j B_j^\top,$$

where $S_j \in \mathbb{R}^{m \times m}$ is the diagonal matrix with $m$-largest singular values of $X_j$ and $A_j \in \mathbb{R}^{N \times m}$ and $B_j \in \mathbb{R}^{m \times d_j}$ are the corresponding left and right singular vectors. Given this SVD, write the $j^{th}$ projection matrix as

$$\tilde{P}_j = A_j S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j A_j^\top,$$

$$= A_j T_j T_j^\top A_j^\top,$$

where $T_j \in \mathbb{R}^{m \times m}$ is a diagonal matrix such that $T_j T_j^\top = S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j$. Finally, we note
that the sum of projection matrices can be expressed as \( M = \tilde{M} \tilde{M}^\top \) where
\[
\tilde{M} = [A_1T_1 \ldots A_JT_J] \in \mathbb{R}^{N \times m_J}.
\]

Therefore, eigenvectors of matrix \( M \), i.e. the matrix \( G \) that we are interested in finding, are the left singular vectors of \( \tilde{M} \), i.e. \( \tilde{M} = GSV^\top \). These left singular vectors can be computed by using Incremental PCA (Brand, 2002) since \( \tilde{M} \) may be too large to fit in memory.

### 3.1 Computing SVD of mean centered \( X_j \)

Recall that we assumed \( X_j \) to be mean centered matrices. Let \( Z_j \in \mathbb{R}^{N \times d_j} \) be sparse matrices containing mean-uncentered cooccurrence counts. Let \( f_j = n_j \circ t_j \) be the preprocessing function that we apply to \( Z_j \):
\[
\begin{align*}
Y_j &= f_j(Z_j), \\
X_j &= Y_j - 1(1^\top Y_j).
\end{align*}
\]

In order to compute the SVD of mean centered matrices \( X_j \) we first compute the partial SVD of uncentered matrix \( Y_j \) and then update it (Brand (2006) provides details). We experimented with representations created from the uncentered matrices \( Y_j \) and found that they performed as well as the mean centered versions but we would not mention them further since it is computationally efficient to follow the principled approach. We note, however, that even the method of mean-centering the SVD produces an approximation.

### 3.2 Handling missing rows across views

With real data it may happen that a term was not observed in a view at all. A large number of missing rows can corrupt the learnt representations since the rows in the left singular matrix become zero. To counter this problem we adopt a variant of the “missing-data passive” algorithm from Van De Velden and Bijnolm (2006) who modified the GCCA objective to counter the problem of missing rows.$^1$ The objective now becomes:
\[
\begin{align*}
\arg \min_{G,U_j} & \sum_{j=1}^J \| K_j(G - X_jU_j) \|_F^2 \quad \text{subject to } G^\top G = I,
\end{align*}
\]

where \( [K_j]_{ii} = 1 \) if row \( i \) of view \( j \) is observed and zero otherwise. Essentially \( K_j \) is a diagonal row-selection matrix which ensures that we optimize our representations only on the observed rows. Note that \( X_j = K_jX_j \) since the rows that \( K_j \) removed were already zero. Let, \( K = \sum_j K_j \) then the optima of the objective can be computed by modifying equation (7) as:
\[
M = K^{-\frac{1}{2}} \left( \sum_j P_j \right) K^{-\frac{1}{2}}.
\]

Again, if we regularize and approximate the GCCA solution we get \( G \) as the left singular vectors of \( K^{-\frac{1}{2}} M \). We mean center the matrices using only the observed rows.

Also note that other heuristic weighting schemes could be used here. For example if we modify our objective as follows then we would approximately recover the objective of Pennington et al. (2014):
\[
\begin{align*}
\minimize_{G,U_j} & \sum_{j=1}^J \| W_j K_j(G - X_jU_j) \|_F^2 \\
\text{subject to: } & G^\top G = I
\end{align*}
\]

where
\[
[W_j]_{ii} = \left( \frac{w_i}{w_{\max}} \right)^\frac{3}{4} \text{ if } w_i < w_{\max} \text{ else } 1,
\]
and \( w_i = \sum_k [X_j]_{ik} \).

### 4 Data

**Training Data** We used the English portion of the *Polyglot* Wikipedia dataset released by Al-Rfou et al.\footnote{A more recent effort, by van de Velden and Takane (2012), describes newer iterative and non-iterative (Test-Equating Method) approaches for handling missing values. It is possible that using one of those methods could improve performance.}
al. (2013) to create 15 **irredundant** views of coocurrence statistics where element $z_{ij}$ of view $Z_k$ represents that number of times word $w_j$ occurred $k$ words behind $w_i$. We selected the top 500K words by occurrence to create our vocabulary for the rest of the paper.

We extracted cooccurrence statistics from a large bitext corpus that was made by combining a number of parallel bilingual corpora as part of the ParaPhrase DataBase (PPDB) project: Table 1 gives a summary, Ganitkevitch et al. (2013) provides further details. Element $z_{ij}$ of the bitext matrix represents the number of times English word $w_i$ was automatically aligned to the foreign word $w_j$.

We also used the dependency relations in the Annotated Gigaword Corpus (Napoles et al., 2012) to create 21 views\(^2\) where element $z_{ij}$ of view $Z_d$ represents the number of times word $w_j$ occurred as the governor of word $w_i$ under dependency relation $d$.

We combined the knowledge of paraphrases present in FrameNet and PPDB by using the dataset created by Rastogi and Van Durme (2014) to construct a FrameNet view. Element $z_{ij}$ of the FrameNet view represents whether word $w_i$ was present in frame $f_j$. Similarly we combined the knowledge of morphology present in the CatVar database released by Habash and Dorr (2003) and morpha released by Minnen et al. (2001) along with morphy that is a part of WordNet. The morphological views and the frame semantic views were especially sparse with densities of 0.0003\% and 0.03\%. While the approach allows for an arbitrary number of distinct sources of semantic information, such as going further to include cocurrence in WordNet synsets, we considered the described views to be representative, with further improvements possible as future work.

**Test Data** We evaluated the representations on the word similarity datasets listed in Table 2. The first 10 datasets in Table 2 were annotated with different rubrics and rated on different scales. But broadly they all contain human judgements about how similar two words are. The “AN-SYN” and “AN-SEM” datasets contain 4-tuples of analogous words and the
dependency relations employed: nsubj, amod, advmod, rcomd, dobj, prep_of, prep_in, prep_to, prep_on, prep_for, prep_with, prep_from, prep_at, prep_by, prep_as, prep_between, xsubj, agent, conj_and, conj_but, pobj.

![Diagram](image.png)

Figure 1: An illustration of datasets used.

<table>
<thead>
<tr>
<th>Language</th>
<th>Sentences</th>
<th>English Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitext-Arabic</td>
<td>8.8M</td>
<td>190M</td>
</tr>
<tr>
<td>Bitext-Czech</td>
<td>7.3M</td>
<td>17M</td>
</tr>
<tr>
<td>Bitext-German</td>
<td>1.8M</td>
<td>44M</td>
</tr>
<tr>
<td>Bitext-Spanish</td>
<td>11.1M</td>
<td>241M</td>
</tr>
<tr>
<td>Bitext-French</td>
<td>30.9M</td>
<td>671M</td>
</tr>
<tr>
<td>Bitext-Chinese</td>
<td>10.3M</td>
<td>215M</td>
</tr>
<tr>
<td>Monotext-En-Wiki</td>
<td>75M</td>
<td>1700M</td>
</tr>
</tbody>
</table>

Table 1: Portion of data used to create GCCA representations (in millions).

The task is to predict the missing word given the first three. Both of these are open vocabulary tasks while TOEFL is a closed vocabulary task.

### 4.1 Significance of comparison

While surveying the literature we found that performance on word similarity datasets is typically reported in terms of the Spearman correlation between the gold ratings and the cosine distance between normalized embeddings. However researchers do not report measures of significance of the difference between the Spearman Correlations even for comparisons on small evaluation sets.\(^3\) This motivated our defining a method for calculating the **Minimum Required Difference for Significance (MRDS)**.

**Minimum Required Difference for Significance (MRDS):** Imagine two lists of ratings over the same

---

\(^2\)Dependency relations employed: nsubj, amod, advmod, rcomd, dobj, prep_of, prep_in, prep_to, prep_on, prep_for, prep_with, prep_from, prep_at, prep_by, prep_as, prep_between, xsubj, agent, conj_and, conj_but, pobj.

\(^3\)For example, the comparative difference by competing algorithms reported by Faruqui et al. (2014) could not be significant for the Word Similarity test set released by Finkelstein et al. (2001), even if we assumed a correlation between competing methods as high as 0.9, with a p value threshold of 0.05. Similar such comparisons on small datasets are performed by Hill et al. (2014a).
Table 2: List of test datasets used. The columns headed $\sigma^r_{p_0}$ contain MRDS values. The rows for accuracy based test sets contain $\sigma_{p_0}$ which does not depend on $r$. See §4.1 for details.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Size</th>
<th>$\sigma^0.5_{0.01}$</th>
<th>$\sigma^0.7_{0.01}$</th>
<th>$\sigma^0.9_{0.01}$</th>
<th>$\sigma^0.5_{0.05}$</th>
<th>$\sigma^0.7_{0.05}$</th>
<th>$\sigma^0.9_{0.05}$</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEN</td>
<td>3000</td>
<td>4.2</td>
<td>3.2</td>
<td>1.8</td>
<td>3.0</td>
<td>2.3</td>
<td>1.3</td>
<td>(Bruni et al., 2012)</td>
</tr>
<tr>
<td>RW</td>
<td>2034</td>
<td>5.1</td>
<td>3.9</td>
<td>2.3</td>
<td>3.6</td>
<td>2.8</td>
<td>1.6</td>
<td>(Luong et al., 2013)</td>
</tr>
<tr>
<td>SCWS</td>
<td>2003</td>
<td>5.1</td>
<td>4.0</td>
<td>2.3</td>
<td>3.6</td>
<td>2.8</td>
<td>1.6</td>
<td>(Huang et al., 2012)</td>
</tr>
<tr>
<td>SIMLEX</td>
<td>999</td>
<td>7.3</td>
<td>5.7</td>
<td>3.2</td>
<td>5.2</td>
<td>4.0</td>
<td>2.3</td>
<td>(Hill et al., 2014b)</td>
</tr>
<tr>
<td>WS</td>
<td>353</td>
<td>12.3</td>
<td>9.5</td>
<td>5.5</td>
<td>8.7</td>
<td>6.7</td>
<td>3.9</td>
<td>(Finkelstein et al., 2001)</td>
</tr>
<tr>
<td>MTURK</td>
<td>287</td>
<td>13.7</td>
<td>10.6</td>
<td>6.1</td>
<td>9.7</td>
<td>7.5</td>
<td>4.3</td>
<td>(Radinsky et al., 2011)</td>
</tr>
<tr>
<td>WS-REL</td>
<td>252</td>
<td>14.6</td>
<td>11.3</td>
<td>6.5</td>
<td>10.3</td>
<td>8.0</td>
<td>4.6</td>
<td>(Agirre et al., 2009)</td>
</tr>
<tr>
<td>WS-SEM</td>
<td>203</td>
<td>16.2</td>
<td>12.6</td>
<td>7.3</td>
<td>11.5</td>
<td>8.9</td>
<td>5.1</td>
<td>-Same-As-Above-</td>
</tr>
<tr>
<td>RG</td>
<td>65</td>
<td>28.6</td>
<td>22.3</td>
<td>12.9</td>
<td>20.6</td>
<td>16.0</td>
<td>9.2</td>
<td>(Rubenstein and Goodenough, 1965)</td>
</tr>
<tr>
<td>MC</td>
<td>30</td>
<td>41.7</td>
<td>32.7</td>
<td>19.0</td>
<td>30.6</td>
<td>23.9</td>
<td>13.8</td>
<td>(Miller and Charles, 1991)</td>
</tr>
<tr>
<td>AN-SYN</td>
<td>10675</td>
<td>-</td>
<td>-</td>
<td>0.95</td>
<td>-</td>
<td>-</td>
<td>0.68</td>
<td>(Mikolov et al., 2013a)</td>
</tr>
<tr>
<td>AN-SEM</td>
<td>8869</td>
<td>-</td>
<td>-</td>
<td>1.03</td>
<td>-</td>
<td>-</td>
<td>0.74</td>
<td>-Same-As-Above-</td>
</tr>
<tr>
<td>TOEFL</td>
<td>80</td>
<td>-</td>
<td>-</td>
<td>8.13</td>
<td>-</td>
<td>-</td>
<td>6.63</td>
<td>(Landauer and Dumais, 1997)</td>
</tr>
</tbody>
</table>

Let $\sigma^r_{p_0}$ denote Steiger’s test predicate which satisfies the following:

$$\text{stest-p}(\hat{r}_{AT}, \hat{r}_{BT}, r_{AB}, p_0, n) \implies \text{pval} > p_0$$

. Here $\text{pval}$ is the probability of the test statistic under the null hypothesis that $r_{AT} = r_{BT}$ found using the Steiger’s test (Steiger, 1980). The above constraint ensures that as long as the correlation between the competing methods is less than $r$ and the difference between the correlations of the scores of the competing methods to the gold ratings is less than $\sigma^r_{p_0}$, then the pvalue of the null hypothesis will be greater than $p_0$. We can then ask what we consider a reasonable upper bound on the agreement of ratings produced by competing algorithms: for instance two algorithms correlating above 0.9 might not be considered meaningfully different. That leaves us with the second part of the predicate which ensures that as long as the difference between the correlations of the competing algorithms to the gold scores is less than $\sigma^r_{p_0}$ then the null hypothesis is more likely than $p_0$.

We can find $\sigma^r_{p_0}$ as follows: Let $\text{stest}$ denote

$$0 < (\hat{\theta}_B - \hat{\theta}_A) < \sigma_{p_0} \implies p(\theta_B \leq \theta_A) > p_0$$

Specifically, we calculated the posterior probability $p(\theta_B \leq \theta_A)$ with a flat prior of $\beta(1, 1)$ to solve the following:

$$\sigma_{p_0} = \min\{\sigma | \forall 0 < \theta < \min(1-\sigma, 0.9) p(\theta_B \leq \theta_A | \hat{\theta}_A = \theta, \hat{\theta}_B = \theta + \sigma, n) < p_0\}$$

Here $\theta_A$ and $\theta_B$.

This instead of using McNemar’s test (McNemar, 1947) since the Bayesian approach is tractable and more direct. A calculation with $\beta(0.5, 0.5)$ as the prior changed $\sigma_{0.5}$ from 6.63 to 6.38 for the TOEFL dataset but did not affect MRDS for the AN-SEM and AN-SYN datasets.
are probability of correctness of algorithms $A$, $B$ and $\hat{\theta}_A$, $\hat{\theta}_B$ are observed empirical accuracies.

Unfortunately there are no widely reported train-test splits of the above datasets, leading to potential concerns of soft supervision (hyper-parameter tuning) on these evaluations, both in our own work and throughout the existing literature. We report on the resulting impact of various parameterizations, and our final results are based on a single set of parameters used across all evaluation sets.

5 Experiments and Results

We wanted to answer the following questions through our experiments: (1) How do hyper-parameters affect performance? (2) What is the contribution of the multiple sources of data to performance? (3) How does the performance of MVLSA compare with other methods? For brevity we show tuning runs only on the larger datasets. We also highlight the top performing configurations in bold using the small threshold values in column $\sigma_{0.05}^{0.09}$.

Effect of Hyper-parameters $f_j$: We modeled the preprocessing function $f_j$ as the composition of two functions, $f_j = n_j \circ t_j$. $n_j$ represents nonlinear preprocessing that is usually employed with LSA. We experimented by setting $n_j$ to be: identity; logarithm of count plus one; and the fourth root of the count. $t_j$ represents the truncation of columns and can be interpreted as a type of regularization of the raw counts themselves through which we prune away the noisy contexts. Decrease in $t_j$ also reduces the influence of views that have a large number of context columns and emphasizes the sparser views. Table 3 and Table 4 show the results.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Log</th>
<th>Count</th>
<th>Count/$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEN</td>
<td>67.5</td>
<td>59.7</td>
<td><strong>70.7</strong></td>
</tr>
<tr>
<td>RW</td>
<td>31.1</td>
<td>25.3</td>
<td><strong>37.8</strong></td>
</tr>
<tr>
<td>SCWS</td>
<td>64.2</td>
<td>58.2</td>
<td><strong>66.6</strong></td>
</tr>
<tr>
<td>AN-SYN</td>
<td>45.7</td>
<td>21.1</td>
<td><strong>53.6</strong></td>
</tr>
<tr>
<td>AN-SEM</td>
<td>25.4</td>
<td>15.9</td>
<td><strong>38.7</strong></td>
</tr>
</tbody>
</table>

Table 3: Performance versus $n_j$, the non linear processing of cooccurrence counts. $t = 200K$, $m = 500$, $v = 16$, $k = 300$. All the top configurations determined by $\sigma_{0.05}^{0.09}$ are in bold font.

$5$: The number of left singular vectors extracted after SVD of the preprocessed cooccurrence matrices can again be interpreted as a type of regularization, since the result of this truncation is that we find cooccurrence patterns only between the top left singular vectors. We set $m_j = \max(d_j, m)$ with $m = [100, 300, 500]$. See table 5.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEN</td>
<td>65.6</td>
<td>68.5</td>
<td><strong>70.1</strong></td>
<td><strong>71.1</strong></td>
</tr>
<tr>
<td>RW</td>
<td>34.6</td>
<td>36.0</td>
<td><strong>37.2</strong></td>
<td><strong>37.1</strong></td>
</tr>
<tr>
<td>SCWS</td>
<td>64.2</td>
<td>65.4</td>
<td><strong>66.4</strong></td>
<td><strong>66.5</strong></td>
</tr>
<tr>
<td>AN-SYN</td>
<td>50.5</td>
<td><strong>56.2</strong></td>
<td><strong>56.4</strong></td>
<td><strong>56.4</strong></td>
</tr>
<tr>
<td>AN-SEM</td>
<td>24.3</td>
<td>31.4</td>
<td>34.3</td>
<td><strong>40.6</strong></td>
</tr>
</tbody>
</table>

Table 5: Performance versus $m$, the number of left singular vectors extracted from raw cooccurrence counts. We set $n_j = \text{Count}^4$, $t = 100K$, $v = 25$, $k = 300$.

$k$: Table 6 demonstrates the variation in performance versus the dimensionality of the learnt vector representations of the words. Since the dimensions of the MVLSA representations are orthogonal to each other therefore creating lower dimensional representations is a trivial matrix slicing operation and does not require retraining.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEN</td>
<td>49.0</td>
<td>67.0</td>
<td><strong>69.7</strong></td>
<td><strong>70.2</strong></td>
<td><strong>70.1</strong></td>
<td>69.8</td>
</tr>
<tr>
<td>RW</td>
<td>28.8</td>
<td>33.3</td>
<td>35.0</td>
<td>35.2</td>
<td><strong>37.2</strong></td>
<td><strong>38.3</strong></td>
</tr>
<tr>
<td>SCWS</td>
<td>57.8</td>
<td>64.4</td>
<td><strong>65.2</strong></td>
<td><strong>66.1</strong></td>
<td><strong>66.4</strong></td>
<td><strong>65.1</strong></td>
</tr>
<tr>
<td>AN-SYN</td>
<td>9.0</td>
<td>41.2</td>
<td>52.2</td>
<td>55.4</td>
<td><strong>56.4</strong></td>
<td>54.4</td>
</tr>
<tr>
<td>AN-SEM</td>
<td>2.5</td>
<td>21.8</td>
<td>34.8</td>
<td><strong>35.8</strong></td>
<td>34.3</td>
<td>33.8</td>
</tr>
</tbody>
</table>

Table 6: Performance versus $k$, the final dimensionality of the embeddings. We set $m = 300$ and other settings were same as Table 5.

$v$: Expression 12 describes a method to set $W_j$. We experimented with a different, more global,
heuristic to set $|W_j|_{ii} = (K_{ww} \geq v)$, essentially removing all words that did not appear in $v$ views before doing GCCA. Table 7 shows that changes in $v$ are largely inconsequential for performance.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>16</th>
<th>17</th>
<th>21</th>
<th>25</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEN</td>
<td>70.4</td>
<td>70.4</td>
<td>70.2</td>
<td>70.1</td>
<td>70.0</td>
</tr>
<tr>
<td>RW</td>
<td>39.9</td>
<td>38.8</td>
<td>39.7</td>
<td>37.2</td>
<td>33.5</td>
</tr>
<tr>
<td>SCWS</td>
<td>67.0</td>
<td>66.8</td>
<td>66.5</td>
<td>66.4</td>
<td>65.7</td>
</tr>
<tr>
<td>AN-SYN</td>
<td>56.0</td>
<td>55.8</td>
<td>55.9</td>
<td>56.4</td>
<td>56.0</td>
</tr>
<tr>
<td>AN-SEM</td>
<td>34.6</td>
<td>34.3</td>
<td>34.0</td>
<td>34.3</td>
<td>34.3</td>
</tr>
</tbody>
</table>

Table 7: Performance versus minimum view support threshold $v$. The other hyperparameters were $n_j = \text{Count}^2$, $m = 300$, $t = 100K$. Though a clear best setting did not emerge, we chose $v = 25$ as the middle ground.

$r_j$: The regularization parameter ensures that all the inverses exist at all points in our method. We found that the performance of our procedure was invariant to $r$ over a large range from 1 to 1e-10. This was because even the 1000th singular value of our data was much higher than 1.

**Contribution of different sources of data** Table 8 shows an ablative analysis of performance where we remove individual views or some combination of them and measure the performance. It is clear by comparing the last column to the second column that adding in more views improves performance. Also we can see that the Dependency based views and the Bitext based views give a larger boost than the morphology and FrameNet based views, probably because the latter are so sparse.

**Comparison to other word representation creation methods** There are a large number of methods of creating representations both multilingual and monolingual. There are many new methods such as by Yu and Dredze (2014), Faruqui et al. (2014), Hill and Korhonen (2014), and Weston et al. (2014) that are performing multiview learning and could be considered here as baselines: however it is not straightforward to use those systems to handle the variety of data that we are using. Therefore, we directly compare our method to the Glove and the SkipGram model of Word2Vec as the performance of those systems is considered state of the art. We trained these two systems on the English portion of the Polyglot Wikipedia dataset. We also combined their outputs using MVLSA to create MV-G-WSG embeddings.

We trained our best MVLSA system with data from all views and by using the individual best settings of the hyper-parameters. Specifically the configuration we used was as follows: $n_j = \text{Count}^2$, $t = 12.5K$, $m = 500$, $k = 300$, $v = 16$. To make a fair comparison we also provide results where we used only the views derived from the Polyglot Wikipedia corpus. See column MVLSA (All Views) and MVLSA (Wiki) respectively. It is clearly visible that MVLSA on the monolingual data itself is competitive with Glove but worse than Word2Vec on the word similarity datasets and it is substantially worse than both the systems on the AN-SYN and AN-SEM datasets. However with the addition of multiple views MVLSA makes substantial gains, shown in column MV Gain, and after consuming the Glove and WSG embeddings it again improves performance by some margins, as shown in column G-WSG Gain, and outperforms the original systems. Using GCCA itself for system combination provides closure for the MVLSA algorithm since multiple distinct approaches can now be simply fused using this method. Finally we contrast the Spearman correlations $r_s$ with Glove and Word2Vec before and after including them in the GCCA procedure. The values demonstrate that including Glove and WSG during GCCA actually increased the correlation between them and the learnt embeddings, which supports our motivation for performing GCCA in the first place.

### 6 Previous Work

Vector space representations of words have been created using diverse frameworks including Spectral methods (Dhillon et al., 2011; Dhillon et al., 2012), Neural Networks (Mikolov et al., 2013b; Collobert and Lebret, 2013), and Random Projections (Ravichandran et al., 2005; Bhagat and Ravichandran).
Table 8: Performance versus views removed from the multiview GCCA procedure. !Framenet means that the view containing counts derived from Frame semantic dataset was removed. Other columns are named similarly. The other hyperparameters were $n_j = \text{Count}^\dagger$, $m = 300$, $t = 100K$, $v = 25$, $k = 300$.

Table 9: Comparison of Multiview LSA against Glove and WSG (Word2Vec Skip Gram). Using $\sigma_{0.95}$ as the threshold we highlighted the top performing systems in bold font. $^\dagger$ marks significant increments in performance due to use of multiple views in the Gain columns. The $r_\sigma$ columns demonstrate that GCCA increased Pearson correlation.

---

7code.google.com/p/word2vecmetaoptimize.com/projects/wordreprs
8nlp.stanford.edu/projects/glove
9ttic.uchicago.edu/~mbansal/data/syntacticEmbeddings.zip, cs.cmu.edu/~mfaruqui/soft.html

---

dran, 2008; Chan et al., 2011). They have been trained using either one (Pennington et al., 2014) or two sources of cooccurrence statistics (Zou et al., 2013; Faruqui and Dyer, 2014; Bansal et al., 2014; Levy and Goldberg, 2014) or using multi-modal data (Hill et al. and Korhonen, 2014; Bruni et al., 2012).

Dhillon et al. (2011) and Dhillon et al. (2012) were the first to use CCA as the primary method to learn vector representations and Faruqui and Dyer (2014) further demonstrated that incorporating bilingual data through CCA improved performance. More recently this same phenomenon was reported by Hill et al. (2014a) through their experiments over neural representations learnt from MT systems. Various other researchers have tried to improve the performance of their paraphrase systems or vector space models by using diverse sources of information such as bilingual corpora (Bannard and Callison-Burch, 2005; Huang et al., 2012; Zou et al., 2013), structured datasets (Yu and Dredze, 2014; Faruqui et al., 2014) or even tagged images (Bruni et al., 2011).
et al., 2012). However, most previous work\textsuperscript{11} did not adopt the general, simplifying view that all of these sources of data are just cooccurrence statistics coming from different sources with underlying latent factors.\textsuperscript{12}

Bach and Jordan (2005) presented a probabilistic interpretation for CCA. Though they did not generalize it to include GCCA we believe that one could give a probabilistic interpretation of MAX-VAR GCCA. Such a probabilistic interpretation would allow for an online-generative model of lexical representations, which unlike methods like Glove or LSA would allow us to naturally perplexity or generate sequences. We also note that Vía et al. (2007) presented a neural network model of GCCA and adaptive/incremental GCCA. To the best of our knowledge both of these approaches have not been used for word representation learning.

CCA is also an algorithm for multi-view learning (Kakade and Foster, 2007; Ganchev et al., 2008) and when we view our work as an application of multi-view learning to NLP, this follows a long chain of effort started by Yarowsky (1995) and continued with Co-Training (Blum and Mitchell, 1998), CoBoosting (Collins and Singer, 1999) and 2 view perceptrons (Brefeld et al., 2006).

7 Conclusion and Future Work

While previous efforts demonstrated that incorporating two views is beneficial in word-representation learning, we extended that thread of work to a logical extreme and created MVLSA to learn distributed representations using data from 46 views\textsuperscript{13}.

Through evaluation of our induced representations, shown in Table 9, we demonstrated that the MVLSA algorithm is able to leverage the information present in multiple data sources to improve performance on a battery of tests against state of the art baselines.

In order to perform MVLSA on large vocabularies with up to 500K words we presented a fast scalable algorithm. We also showed that a close variant of the Glove objective proposed by Pennington et al. (2014) could be derived as a heuristic for handling missing data under the MVLSA framework. In order to better understand the benefit of using multiple sources of data we performed MVLSA using views derived only from the monolingual Wikipedia dataset thereby providing a more principled alternative of LSA that removes the need for heuristically combining word-word cooccurrence matrices into a single matrix. Finally, while surveying the literature we noticed that not enough emphasis was being given towards establishing the significance of comparative results and proposed a method, (MRDS), to filter out insignificant comparative gains between competing algorithms.

Future Work

Column MVLSA Wiki of Table 9 shows us that MVLSA applied to monolingual data has mediocre performance compared to the baselines of Glove and Word2Vec on word similarity tasks and performs surprisingly worse on the ANSEM dataset. We believe that the results could be improved by (1) either using recent methods for handling missing values mentioned in footnote 1 or by using the heuristic count dependent non-linear weighting mentioned by Pennington et al. (2014) and that sits well within our framework as exemplified in Expression 12 (2) by using even more views, which look at the future words as well as views that contain PMI values. Finally, we note that Table 8 shows that certain datasets can actually degrade performance over certain metrics. Therefore we are exploring methods for performing discriminative optimization of weights assigned to views, for purposes of task-based customization of learned representations.

Acknowledgments

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under the Deep Exploration and Filtering of Text (DEFT) Program, agreement number FA8750-13-2-001, as well as the National Science Foundation (NSF), agreement number BCS-1344269. We also thank Juri Ganitkevitch for providing the word aligned bitext corpus.
References


Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. JMLR, 15.


Paramveer Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via CCA. In Proceedings of NIPS.


Felix Hill and Anna Korhonen. 2014. Learning abstract concept embeddings from multi-modal data: Since you probably can’t see what i mean. In Proceedings of EMNLP. ACL.


Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of ACL. ACL.


Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In Proceedings of ACL. ACL.


Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2).


Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In Proceedings of ACL. ACL.