Deep Multilingual Correlation for Improved Word Embeddings

Ang Lu\textsuperscript{1}, Weiran Wang\textsuperscript{2}, Mohit Bansal\textsuperscript{2}, Kevin Gimpel\textsuperscript{2}, and Karen Livescu\textsuperscript{2}

\textsuperscript{1}Department of Automation, Tsinghua University, Beijing, 100084, China
lval1@mails.tsinghua.edu.cn
\textsuperscript{2}Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA
\{weiranwang, mbansal, kgimpel, klivescu\}@ttic.edu

Abstract

Word embeddings have been found useful for many NLP tasks, including part-of-speech tagging, named entity recognition, and parsing. Adding multilingual context when learning embeddings can improve their quality, for example via canonical correlation analysis (CCA) on embeddings from two languages. In this paper, we extend this idea to learn deep non-linear transformations of word embeddings of the two languages, using the recently proposed deep canonical correlation analysis. The resulting embeddings, when evaluated on multiple word and bigram similarity tasks, consistently improve over monolingual embeddings and over embeddings transformed with linear CCA.

1 Introduction

Learned word representations are widely used in NLP tasks such as tagging, named entity recognition, and parsing (Miller et al., 2004; Koo et al., 2008; Turian et al., 2010; Täckström et al., 2012; Huang et al., 2014; Bansal et al., 2014). The idea in such representations is that words with similar context have similar meaning, and hence should be nearby in a clustering or vector space. Continuous representations are learned with neural language models (Bengio et al., 2003; Mnih and Hinton, 2007; Mikolov et al., 2013) or spectral methods (Deerwester et al., 1990; Dhillon et al., 2011). The context used to learn these representations is typically the set of nearby words of each word occurrence. Prior work has found that adding translational context results in better representations (Diab and Resnik, 2002; Täckström et al., 2012; Bansal et al., 2012; Zou et al., 2013). Recently, Faruqui and Dyer (2014) applied canonical correlation analysis (CCA) to word embeddings of two languages, and found that the resulting embeddings represent word similarities better than the original monolingual embeddings.

In this paper, we follow the same intuition as Faruqui and Dyer (2014) but rather than learning linear transformations with CCA, we permit the correlated information to lie in nonlinear subspaces of the original embeddings. We use the recently proposed deep canonical correlation analysis (DCCA) technique of Andrew et al. (2013) to learn nonlinear transformations of two languages’ embeddings that are highly correlated. We evaluate our DCCA-transformed embeddings on word similarity tasks like WordSim-353 (Finkelstein et al., 2001) and SimLex-999 (Hill et al., 2014), and also on the bigram similarity task of Mitchell and Lapata (2010) (using additive composition), obtaining consistent improvements over the original embeddings and over linear CCA. We also compare tuning criteria and ensemble methods for these architectures.

2 Method

We assume that we have initial word embeddings for two languages, denoted by random vectors $x \in \mathbb{R}^{D_x}$ and $y \in \mathbb{R}^{D_y}$, and a set of bilingual word pairs. Our goal is to obtain a representation for each language that incorporates useful information from both $x$ and $y$. We consider the two input monolingual word embeddings as different views of the same latent semantic signal. There are multiple ways to incorporate multilingual information into word embeddings. Here we follow Faruqui and Dyer (2014) in taking a CCA-based approach, in which we project the original embeddings onto their maximally correlated subspaces. However, instead of relying on linear correlation, we learn more powerful non-linear transformations of each view via deep networks.

Canonical Correlation Analysis A popular method for multi-view representation learning is canonical correlation analysis (CCA; Hotelling, 1936). Its objective is to find two vectors $u \in \mathbb{R}^{D_x}$ and $v \in \mathbb{R}^{D_y}$ such that the covariance between $u$ and $v$ is maximized, subject to constraints on the variances of $u$ and $v$.
and \( \mathbf{v} \in \mathbb{R}^{D_y} \) such that projections of the two views onto these vectors are maximally (linearly) correlated:

\[
\max_{\mathbf{u} \in \mathbb{R}^{D_x}, \mathbf{v} \in \mathbb{R}^{D_y}} \frac{\mathbb{E} \left[ (\mathbf{u}^\top \mathbf{x})(\mathbf{v}^\top \mathbf{y}) \right]}{\sqrt{\mathbb{E}[(\mathbf{u}^\top \mathbf{x})^2]} \sqrt{\mathbb{E}[(\mathbf{v}^\top \mathbf{y})^2]}} = \frac{\mathbf{u}^\top \Sigma_{xy} \mathbf{v}}{\sqrt{\mathbf{u}^\top \Sigma_{xu} \mathbf{u}} \sqrt{\mathbf{v}^\top \Sigma_{yy} \mathbf{v}}} = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} x_i y_i
\]

which allows: \( \Sigma_{xy} \) and \( \Sigma_{xx} \) are the cross-view and within-view covariance matrices. (1) is extended to learn multi-dimensional projections by optimizing the sum of correlations in all dimensions, subject to different projected dimensions being uncorrelated. Given sample pairs \( \{(x_i, y_i)\}_{i=1}^{N} \), the empirical estimates of the covariance matrices are

\[
\hat{\Sigma}_{xx} = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^\top + r_x I, \quad \hat{\Sigma}_{yy} = \frac{1}{N} \sum_{i=1}^{N} y_i y_i^\top + r_y I \quad \text{and} \quad \hat{\Sigma}_{xy} = \frac{1}{N} \sum_{i=1}^{N} x_i y_i^\top
\]

where \( (r_x, r_y) > 0 \) are regularization parameters (Hardoon et al., 2004; De Bie and De Moor, 2003). Then the optimal \( k \)-dimensional projection mappings are given in closed form via the rank-\( k \) singular value decomposition (SVD) of the \( D_x \times D_y \) matrix \( \hat{\Sigma}_{xx}^{-1/2} \hat{\Sigma}_{xy} \hat{\Sigma}_{yy}^{-1/2} \).

### 2.1 Deep Canonical Correlation Analysis

A linear feature mapping is often not sufficiently powerful to faithfully capture the hidden, non-linear relationships within the data. Recently, Andrew et al. (2013) proposed a nonlinear extension of CCA using deep neural networks, dubbed deep canonical correlation analysis (DCCA) and illustrated in Figure 1. In this model, two (possibly deep) neural networks \( f \) and \( g \) are used to extract features from each view, and trained to maximize the correlations between outputs in the two views, measured by a linear CCA step with projection mappings \( (\mathbf{u}, \mathbf{v}) \). The neural network weights and the linear projections are optimized together using the objective

\[
\max_{W_f, W_g, u, v} \frac{\mathbf{u}^\top \Sigma_{fg} \mathbf{v}}{\sqrt{\mathbf{u}^\top \Sigma_{ff} \mathbf{u}} \sqrt{\mathbf{v}^\top \Sigma_{gg} \mathbf{v}}},
\]

where \( W_f \) and \( W_g \) are the weights of the two networks and \( \Sigma_{fg}, \Sigma_{ff} \) and \( \Sigma_{gg} \) are covariance matrices computed for \( \{(f(x_i), g(y_i))\}_{i=1}^{N} \) in the same way as CCA. The final transformation is the composition of the neural network and CCA projection, e.g., \( \mathbf{u}^\top f(x) \) for the first view. Unlike CCA, DCCA does not have a closed-form solution, but the parameters can be learned via gradient-based optimization, with either batch algorithms like L-BFGS as in (Andrew et al., 2013) or a mini-batch stochastic gradient descent-like approach as in (Wang et al., 2015). We choose the latter in this paper.

An alternative nonlinear extension of CCA is kernel CCA (KCCA) (Lai and Fyfe, 2000; Vinokourov et al., 2003), which introduces nonlinearity through kernels. DCCA scales better with data size, as KCCA involves the SVD of an \( N \times N \) matrix. Andrew et al. (2013) showed that DCCA achieves better correlation on held-out data than CCA/KCCA, and Wang et al. (2015) found that DCCA outperforms CCA and KCCA on a speech recognition task.

### 3 Experiments

We use English and German as our two languages. Our original monolingual word vectors are the same as those used by Faruqui and Dyer (2014). They are 640-dimensional and are estimated via latent semantic analysis on the WMT 2011 monolingual news corpora.\(^1\) We use German-English translation pairs as the input to CCA and DCCA, using the same set of 36K pairs as used by Faruqui and Dyer. These pairs contain, for each of 36K English word types, the single most frequently aligned German word. They were obtained using the word aligner in cdec (Dyer et al., 2010) run on the WMT06-10 news commentary corpora and Europarl. After training, we apply the learned CCA/DCCA projection mappings to the original English word embeddings (180K words) and use these transformed embeddings for our evaluation tasks.

### 3.1 Evaluation Tasks

We compare our DCCA-based embeddings to the original word vectors and to CCA-based em-

\(^1\)www.statmt.org/wmt11/
beddings on several tasks. We use WordSim-353 (Finkelstein et al., 2001), which contains 353 English word pairs with human similarity ratings. It is divided into WS-SIM and WS-REL by Agirre et al. (2009) to measure similarity and relatedness. We also use SimLex-999 (Hill et al., 2014), a new similarity-focused dataset consisting of 666 noun pairs, 222 verb pairs, and 111 adjective pairs. Finally, we use the bigram similarity dataset from Mitchell and Lapata (2010) which has 3 subsets, adjective-noun (AN), noun-noun (NN), and verb-object (VN), and dev and test sets for each. For the bigram task, we simply add the word vectors output by CCA or DCCA to get bigram vectors.\(^2\)

All task datasets contain pairs with human similarity ratings. To evaluate embeddings, we compute cosine similarity between the two vectors in each pair, order the pairs by similarity, and compute Spearman’s correlation (\(\rho\)) between the model’s ranking and human ranking.

### 3.2 Training

We normalize the 36K training pair vectors to unit norm (as also done by Faruqui and Dyer). We then remove the per-dimension mean and standard deviation of this set of training pairs, as is typically done in neural network training (LeCun et al., 1998). We do the same to the original 180K English word vectors (normalize to unit norm, remove the mean/standard deviation of the size-36K training set), then apply our CCA/DCCA mappings to these 180K vectors. The resulting 180K vectors are further normalized to zero mean before cosine similarities between test pairs are computed, as also done by Faruqui and Dyer.

For both CCA and DCCA, we tune the output dimensionality among factors in \{0.2, 0.4, 0.6, 0.8, 1.0\} of the original embedding dimension (640), and regularization \((r_x, r_y)\) from \{\(10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}\), based on the 7 tuning tasks discussed below.

For DCCA, we use standard deep neural networks with rectified linear units and tune the depth (1 to 4 hidden layers) and layer widths (in \{128, 256, 512, 1024, 2048, 4096\}) separately for each language. For optimization, we use stochastic gradient descent (SGD) as described by Wang et al. (2015). We tune SGD hyperparameters on a small grid, choosing a mini-batch size of 3000, learning rate of 0.0001, and momentum of 0.99.

### 3.3 Tuning

Our main results are based on tuning hyperparameters (of CCA/DCCA) on 7 word similarity tasks.\(^3\) We perform additional experiments in which we tune on the development sets for the bigram tasks. We set aside WS-353, SimLex-999, and the test sets of the bigram tasks as held-out test sets. We consider two tuning criteria:

- **BestAvg**: Choose the hyperparameters with the best average performance across the 7 tuning tasks. This is the only tuning criterion used for CCA.
- **MostBeat**: For DCCA, choose the hyperparameters that beat the best CCA embeddings on a maximum number of the 7 tasks; to break ties here, choose the hyperparameters with the best average performance. The idea is that we want to find a setting that generalizes to many tasks.

We also consider simple ensembles by averaging the cosine similarities from the three best settings under each of these two criteria.

### 3.4 Results

Table 1 shows our main results on the word and bigram similarity tasks. All values are Spearman’s correlation (\(\rho\)). We show the original word vector results, the best-tuned CCA setting (CCA-1), the ensemble of the top-3 CCA settings (CCA-Ens), and the same for DCCA (with both tuning criteria). The DCCA results show an overall improvement on most tasks over linear CCA (all of the shaded DCCA results are better than all corresponding CCA results). Each of our tuning criteria for DCCA performs well, and almost always better than CCA. BestAvg is better on some tasks while MostBeat is better on others; we report both here to bring attention to and promote discussion about the effects of tuning methods when learning representations in the absence of supervision or in-domain tuning data.

In Table 2, we report additional bigram similarity results obtained by tuning on the dev sets of the bi-

\(^2\)We also tried multiplication but it performed worse. In future work, we will directly train on bigram translation pairs.

\(^3\)RG-65 (Rubenstein and Goodenough, 1965), MC-30 (Miller and Charles, 1991), MTurk-287 (Radinsky et al., 2011), MTurk-771, MEN (Bruni et al., 2014), Rare Word (Luong et al., 2013), and YP-130 (Yang and Powers, 2006).
Table 1: Main results on word and bigram similarity tasks, tuned on 7 development tasks (see text for details). Shading indicates a result that matches or improves the best linear CCA result; boldface indicates the best result in a given column. See Section 3.4 for discussion on NN results.

<table>
<thead>
<tr>
<th>Embeddings</th>
<th>WS-353</th>
<th>WS-SIM</th>
<th>WS-REL</th>
<th>SL-999</th>
<th>AN</th>
<th>NN</th>
<th>VN</th>
<th>Avg</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>46.7</td>
<td>56.3</td>
<td>36.6</td>
<td>26.5</td>
<td>26.5</td>
<td>38.1</td>
<td>34.1</td>
<td>32.9</td>
<td>640</td>
</tr>
<tr>
<td>CCA-1</td>
<td>67.2</td>
<td>73.0</td>
<td>63.4</td>
<td>40.7</td>
<td>42.4</td>
<td>48.1</td>
<td>37.4</td>
<td>42.6</td>
<td>384</td>
</tr>
<tr>
<td>CCA-Ens</td>
<td>67.5</td>
<td>73.1</td>
<td>63.7</td>
<td>40.4</td>
<td>42.0</td>
<td>48.2</td>
<td>37.8</td>
<td>42.7</td>
<td>384</td>
</tr>
<tr>
<td>DCCA-1 (BestAvg)</td>
<td>69.6</td>
<td>73.9</td>
<td>65.6</td>
<td>38.9</td>
<td>35.0</td>
<td>40.9</td>
<td>41.3</td>
<td>39.1</td>
<td>128</td>
</tr>
<tr>
<td>DCCA-Ens (BestAvg)</td>
<td>70.8</td>
<td>75.2</td>
<td>67.3</td>
<td>41.7</td>
<td>42.4</td>
<td>45.7</td>
<td>40.1</td>
<td>42.7</td>
<td>128</td>
</tr>
<tr>
<td>DCCA-1 (MostBeat)</td>
<td>68.6</td>
<td>73.5</td>
<td>65.7</td>
<td>42.3</td>
<td>44.4</td>
<td>44.7</td>
<td>36.7</td>
<td>41.9</td>
<td>384</td>
</tr>
<tr>
<td>DCCA-Ens (MostBeat)</td>
<td>69.9</td>
<td>74.4</td>
<td>66.7</td>
<td>42.3</td>
<td>43.7</td>
<td>47.4</td>
<td>38.8</td>
<td>43.3</td>
<td>384</td>
</tr>
</tbody>
</table>

Table 2: Bigram results, tuned on bigram dev sets.

<table>
<thead>
<tr>
<th>Embeddings</th>
<th>AN</th>
<th>NN</th>
<th>VN</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA</td>
<td>42.4</td>
<td>48.1</td>
<td>37.4</td>
<td>42.6</td>
</tr>
<tr>
<td>Deep CCA</td>
<td>45.5</td>
<td>47.1</td>
<td>45.1</td>
<td>45.9</td>
</tr>
</tbody>
</table>

Table 3: Highly-similar pairs in SimLex-999 that improved/degraded the most under DCCA. Pairs are sorted in decreasing order according to the amount of improvement/degradation.

<table>
<thead>
<tr>
<th>better with DCCA</th>
<th>worse with DCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrive</td>
<td>come</td>
</tr>
<tr>
<td>locate</td>
<td>find</td>
</tr>
<tr>
<td>way</td>
<td>manner</td>
</tr>
<tr>
<td>recent</td>
<td>new</td>
</tr>
<tr>
<td>take</td>
<td>obtain</td>
</tr>
<tr>
<td>boundary</td>
<td>border</td>
</tr>
<tr>
<td>win</td>
<td>accomplish</td>
</tr>
<tr>
<td>contemplate</td>
<td>think</td>
</tr>
</tbody>
</table>

4 Discussion

Normalization and Evaluation We note that the cosine similarity (and thus Spearman’s $\rho$) between a pair of words is not invariant to the series of simple (affine) transformations done by the normalizations in our procedure. For their baseline, Faruqui and Dyer (2014) did not remove the standard deviation of the 36K training set for the 180K English vocabulary during testing. We have accidentally found that this normalization step alone greatly improves the performance of the original vectors.

For example, the WS-353 correlation improves from 46.7 to 67.1, essentially matching the linear CCA correlations, though DCCA still outperforms them both. This indicates that the cosine similarity is not stable, and it is likely better to learn a distance/similarity function (using labeled tuning data) atop the learned features such that similarities between selected pairs will match the human similarities, or such that the rankings will match.

Error Analysis We analyze high-similarity word pairs that change the most with DCCA, as compared to both linear CCA and the original vectors.

For a word pair $w$, we use $r(w)$ to refer to its similarity rank, subscripting it whether it is computed according to human ratings ($r_h$) or if based on cosine similarity via the original vectors ($r_o$), CCA-1 ($r_c$), or DCCA-1 MostBeat ($r_d$). We define $\delta_a(w) = |r_a(w) - r_h(w)|$ and compute $\Delta(w) =$
δ_d(w) − (δ_c(w) + δ_o(w)). If Δ(w) < 0, then word pair w was closer to the human ranking using DCCA. Table 3 shows word pairs from SimLex-999 with high human similarity ratings (≥ 7 out of 10); column 1 shows pairs with smallest Δ values, and column 2 shows pairs with largest Δ values.

Among pairs in column 1, many contain words with several senses. Using bilingual information is likely to focus on the most frequent sense in the bi-text, due to our use of the most frequently-aligned German word in each training pair. By contrast, using only monolingual context is expected to find an embedding that blends the contextual information across all word senses.

Several pairs from column 2 show hypernym rather than paraphrase relationships, e.g., author-creator and leader-manager. Though these pairs are rated as highly similar by annotators, linear CCA made them less similar than the original vectors, and DCCA made them less similar still. This matches our intuition that bilingual information should encourage paraphrase-like similarity and thereby discourage the similarity of hypernym-hyponym pairs.

Visualizations We visualized several synonym-antonym word lists and often found that DCCA more cleanly separated synonyms from antonyms than CCA or the original vectors. An example of the clearest improvement is shown in Fig. 2.

5 Related work

Previous work has successfully used translational context for word representations (Diab and Resnik, 2002; Zhao et al., 2005; Täckström et al., 2012; Bansal et al., 2012; Faruqui and Dyer, 2014), including via hand-designed vector space models (Peirson and Padó, 2010; Sumita, 2000) or via unsupervised LDA and LSA (Boyd-Graber and Blei, 2009; Zhao and Xing, 2006).

There have been other recent deep learning approaches to bilingual representations, e.g., based on a joint monolingual and bilingual objective (Zou et al., 2013). There has also been recent interest in learning bilingual representations without using word alignments (Chandar et al., 2014; Gouws et al., 2014; Kočiský et al., 2014; Vulic and Moens, 2013).

This research is also related to early examples of learning bilingual lexicons using monolingual corpora (Koehn and Knight, 2002; Haghhighi et al., 2008); the latter used CCA to find matched word pairs. Irvine and Callison-Burch (2013) used a supervised learning method with multiple monolingual signals. Finally, other work on CCA and spectral methods has been used in the context of other types of views (Collobert and Weston, 2008; Dhillon et al., 2011; Klementiev et al., 2012; Chang et al., 2013).

6 Conclusion

We have demonstrated how bilingual information can be incorporated into word embeddings via deep canonical correlation analysis (DCCA). The DCCA embeddings consistently outperform linear CCA embeddings on word and bigram similarity tasks. Future work could compare DCCA to other non-linear approaches discussed in §5, compare different languages as multiview context, and extend to aligned phrase pairs, and to unaligned data.

Acknowledgments

We are grateful to Manaal Faruqui for sharing resources, and to Chris Dyer, David Sontag, Lyle Ungar, and anonymous reviewers for helpful input.
References


Ivan Vulic and Marie-Francine Moens. 2013. A study on bootstrapping bilingual vector spaces from non-parallel data (and nothing else). In Proceedings of EMNLP.