Translation Invariant Word Embeddings

Matt Gardner[*]
Carnegie Mellon University
mg1@cs.cmu.edu

Kejun Huang[*]
University of Minnesota
huang663@umn.edu

Evangelos Papalexakis
Carnegie Mellon University
epapalex@cs.cmu.edu

Xiao Fu
University of Minnesota
xfu@umn.edu

Partha Talukdar
Indian Institute of Science
ppt@serc.iisc.in

Christos Faloutsos
Carnegie Mellon University
christos@cs.cmu.edu

Nicholas Sidiropoulos
University of Minnesota
nikos@umn.edu

Tom Mitchell
Carnegie Mellon University
tom.mitchell@cmu.edu

Abstract

This work focuses on the task of finding latent vector representations of the words in a corpus. In particular, we address the issue of what to do when there are multiple languages in the corpus. Prior work has, among other techniques, used canonical correlation analysis to project pre-trained vectors in two languages into a common space. We propose a simple and scalable method that is inspired by the notion that the learned vector representations should be invariant to translation between languages. We show empirically that our method outperforms prior work on multilingual tasks, matches the performance of prior work on monolingual tasks, and scales linearly with the size of the input data (and thus the number of languages being embedded).

1 Introduction

Representing words as vectors in some latent space has long been a central idea in natural language processing. The *distributional hypothesis*, perhaps best stated as “You shall know a word by the company it keeps” (Firth, 1957), has had a long and productive history, as well as a recent revival in neural-network-based models (Mikolov et al., 2013). These methods generally construct a word by context matrix, then either use the vectors directly (often weighted by term frequency and inverse document frequency), perform some factorization of the matrix, or use it as input to a neural network which produces vectors for each word. The resultant vectors can be used in a wide array of tasks, from information retrieval to part-of-speech tagging and parsing.

There has also been some recent work addressing how to create these vectors when information from multiple languages is available. Two recent attempts involve using canonical correlation analysis (CCA) to project pre-trained vectors from each of two languages into a common space (Faruqui and Dyer, 2014b) and using an alignment matrix to heuristically project the vectors from one language onto the words in another language (Guo et al., 2015). These methods generally only work with two languages at a time, however.

In this paper, we introduce a technique for constructing multilingual word embeddings that is inspired by the notion of translational invariance. CCA and the heuristic projection mentioned above both attempt to construct vectors such that words that are translations of each other are close in the vector space, but the method we introduce formalizes this as part of the objective function of the *original decomposition*. We further show how to optimize this objective function with a method that scales linearly in the size of the input data. This results in a scalable, single-step method that is informed by both the monolingual corpus statistics and the multilingual alignment data. We show experimentally that this results in vectors that outperform prior work on multilingual tasks and match the performance of prior work on monolingual tasks.

[*] These authors contributed equally.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1084–1088,
Lisbon, Portugal, 17-21 September 2015.

The contributions of this paper are the following:

- **Problem formulation:** we formalize the notion of translation-invariance, *regardless of the number of languages*, as part of the objective function of a standard matrix decomposition;
- **Scalable algorithm:** we introduce scalable means of optimizing this augmented objective functions; and
- **Effectiveness:** we present state-of-the-art results on a multilingual task using the vectors obtained by these methods.

The code and data used in this paper are publicly available at `https://sites.google.com/a/umn.edu/huang663/`.

2 Problem Definition

The informal problem definition is the following:

Informal Problem. Given *a set of cooccurrence statistics between words in each of several languages, and a translation table containing alignment counts between words in each of these languages,* **Find** *a latent representation for each word in each language that (1) captures information from the cooccurrence statistics and (2) is* invariant *to translations of the cooccurrence statistics between languages.*

More formally, suppose we have M_1 words and N_1 contexts in the first language ("English"), and M_2 and N_2 for the second language ("Spanish"). Then, we are given two matrices of cooccurrence statistics (one for each language), with dimensions $M_1 \times N_1$ and $M_2 \times N_2$, and two dictionary matrices containing translations from English to Spanish, and from Spanish to English, repesctively. A more detailed description on how the data is obtained can be found in (Faruqui and Dyer, 2014b). For simplicity in what follows, we denote these matrices as

- $\mathbf{X}$: a single multilingual cooccurrence matrix (with all the $M_1 + M_2$ words as the rows, and $N_1 + N_2$ contexts as columns). Entries in this matrix specify the cooccurrence between a word in any language and a context in any language.
- $\mathbf{D}_1$: a word dictionary matrix (with all the $M_1 + M_2$ English and Spanish words as both rows and columns). Entries in this matrix specify which words are translations of which other words, and is generally block-normalized, so that (e.g.) each Spanish word has a probability distribution over English words.
- $\mathbf{D}_2$: a context dictionary matrix (with all the $N_1 + N_2$ English and Spanish contexts as both rows and columns). This is similar to $\mathbf{D}_1$ in its construction.

We seek decompositions of $\mathbf{X}$ that are invariant to multiplications along each mode by its respective $\mathbf{D}$ matrix. Note that, while we only described the case where we have two languages, it is straightforward to extend this to having many languages in the combined $\mathbf{X}$, $\mathbf{D}_1$ and $\mathbf{D}_2$ matrices, and we do this in some of the experiments described below.

3 Translation-invariant LSA

Without the side information provided by the dictionary matrices, the classic method for generating word vectors finds a low-rank decomposition of the data matrix $\mathbf{X}$:

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2.$$

With proper scaling (see our discussion in §4.2), the rows of $\mathbf{U}$ (or rows of $\mathbf{V}$) are the word embeddings (or "context embeddings"). It is well-known that the solution is given by the principal components of the singular value decomposition (SVD) of $\mathbf{X}$. Generating word embeddings in this way is known as latent semantic analysis (LSA) (Deerwester et al., 1990).

Our method extends LSA to incorporate information from many languages at a time, with the constraint that the decomposition should be invariant to translation between these languages. We call this method *translation-invariant LSA* (TI-LSA).

In order to take the dictionary matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ into consideration, we propose to seek a decomposition that can simultaneously explain the original matrix $\mathbf{X}$ and various translations of it. We can formalize this in the following objective function:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \; & \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \|\mathbf{D}_1\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \\ & \|\mathbf{X}\mathbf{D}_2^T - \mathbf{U}\mathbf{V}^T\|_F^2 + \|\mathbf{D}_1\mathbf{X}\mathbf{D}_2^T - \mathbf{U}\mathbf{V}^T\|_F^2.\end{aligned} \tag{1}$$

By expanding and combining all four quadratic terms, we can see that the above problem is equivalent to (up to a constant difference)

$$\min_{\mathbf{U},\mathbf{V}} \; \|\tilde{\mathbf{X}} - \mathbf{U}\mathbf{V}^T\|_F^2, \tag{2}$$

where

$$\begin{aligned}\tilde{\mathbf{X}} &= \frac{1}{4}\left(\mathbf{X} + \mathbf{D}_1\mathbf{X} + \mathbf{X}\mathbf{D}_2^T + \mathbf{D}_1\mathbf{X}\mathbf{D}_2^T\right) \\ &= \frac{1}{4}(\mathbf{I} + \mathbf{D}_1)\mathbf{X}(\mathbf{I} + \mathbf{D}_2)^T.\end{aligned}$$

Taking the SVD of $\tilde{\mathbf{X}}$ does not seem numerically appealing at first glance: even though $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{X}$ are all very sparse, forming $\tilde{\mathbf{X}}$ explicitly will introduce a significant amount of nonzeros. However, as we will explain below, it is not necessary to explicitly form $\tilde{\mathbf{X}}$ in order to find a few principal components of it.

We propose to use the Lanczos algorithm (Golub and Van Loan, 1996, Chapter 9) to calculate the SVD of $\tilde{\mathbf{X}}$. The Lanczos method can be viewed as a generalization of the power method for computing an arbitrary number of principal components, and the basic operation required is only matrix-vector multiplication. For our problem specifically, the required matrix-vector multiplications $\tilde{\mathbf{X}}\boldsymbol{\mu}$ and $\tilde{\mathbf{X}}^T\boldsymbol{\nu}$ can be carried out very efficiently with three sparse matrix-vector multiplications, each with complexity linear in the number of nonzeros in the sparse matrix involved, so that any dense intermediate matrix is avoided. As a result, by using our implementation of the Lanczos method, the time required for calculating the SVD of $\tilde{\mathbf{X}}$ is not much more than that of $\mathbf{X}$, even though $\tilde{\mathbf{X}}$ is significantly denser than $\mathbf{X}$.

4 Experiments

We present three experiments to evaluate the method introduced in this paper. The first experiment uses our word embeddings in a cross-lingual dependency parsing task; the second experiment looks at monolingual (English) performance on a series of word-similarity tasks; and the final experiment shows the scalability of our method by applying it to multiple languages.

4.1 Cross-lingual evaluation

Guo et al. (2015) recently introduced a method for using multilingual word embeddings to perform cross-lingual dependency parsing. They train a neural-network-based dependency parsing model using word vectors from one language, and then test the model using data and word vectors from another language. They used the embeddings obtained by Faruqui and Dyer (2014b), along with a heuristic projection. Because we used the same data to obtain our embeddings, our method is directly comparable to the CCA method of Faruqui and Dyer, and the projection method of Guo et al.

Embedding method	LAS	UAS
CCA (Faruqui & Dyer)	60.7	69.8
Projection (Guo et al.)	61.3	71.1
TI-LSA	**62.8**	**72.5**

Table 1: Labeled and unlabeled attachment score (LAS/UAS) on a cross-lingual dependency task. TI-LSA outperforms prior work on this task.

We used code and data graciously provided by Guo to run experiments, training a dependency parsing model on their English treebank, and testing it on the Spanish treebank. We report the results below for the methods used by Guo et al. and the method introduced in this paper. We could not exactly reproduce Guo's result with the code we were provided, so we report all results from our use of the provided code, in case some parameter settings are different from those used in Guo's paper. The results are shown in Table 1. As can be seen in the table, our first method for obtaining multilingual embeddings outperforms both the CCA method of Faruqui and Dyer, and the heuristic projection used by Guo et al.

4.2 Monolingual evaluation

While our focus is on generating embeddings that are invariant to translations (and thus most suited to multi- or cross-lingual tasks), we would hope that the addition of multiple languages would not hurt performance on monolingual tasks. We used wordvectors.org (Faruqui and Dyer, 2014a) to evaluate our learned vectors on a variety of English-language word similarity tasks. The tasks are mostly all variations on performing word similarity judgments, finding the correlation between the system's output and human responses. We used the same data as that used by Faruqui and Dyer (2014b) (English-Spanish only), and thus our method for obtaining multilingual embeddings is directly comparable to their technique for doing the same (CCA). We used the first 11 tasks on wordvectors.org, and obtained Faruqui and Dyer's results from that website. Due to space constraints, we only report the average performance across these 11 tasks for each of the methods we tested. The results are shown in Table 2. To test statistical significance, we performed a paired permutation test, treating performance on each task as

Method	Average Correlation
CCA (Faruqui & Dyer)	0.638
LSA	0.626
TI-LSA	0.628

Table 2: Average correlation with human similarity judgments on 11 word-similarity tasks. The differences between these methods are not statistically significant, showing that the gains we see in cross-lingual tasks are not at the expense of monolingual tasks.

paired data. The important thing to note from the table is that the differences between the methods are all quite small, and none of them are statistically significant.

Note that LSA on just the *English* data performs on par with all of the other methods presented; we have not found a way to improve performance on this monolingual task from using multilingual data.[1] However, it is also important to note that our multilingual methods do not *hurt* performance on these monolingual tasks, either—we get the benefits described in our other evaluations without losing performance on English-only tasks.

4.3 Scalability

We mentioned in Section 3 that our method is linear in the number of nonzeros in the data, as we are simply using the Lanczos algorithm to compute a sparse SVD. To show this in practice, we briefly present how the running time of our algorithm scales with the number of languages used. Each additional language adds roughly the same amount of data to the **X** matrix. Figure 1 shows that our method does indeed scale linearly with the number of nonzeros in the matrix, and thus also with the number of languages used (assuming each language has roughly the same amount of data). All the experiments are performed in MATLAB 2013a on a Linux server with 32 Xeon 2.00GHz cores and 128GB memory.

5 Discussion

We discuss here two points on the flexibility of the method we have introduced. First, note that the dictionary matrices we used contained information about translations between languages. It is also possible to include information about *paraphrases* in this dictionary. For instance, a resource such as the Paraphrase Database (Ganitkevitch et al., 2013) could be used to further constrain the embeddings obtained; this could be useful if the resource used to obtain a paraphrase dictionary contained more or different information than the corpus statistics used in the decomposition.

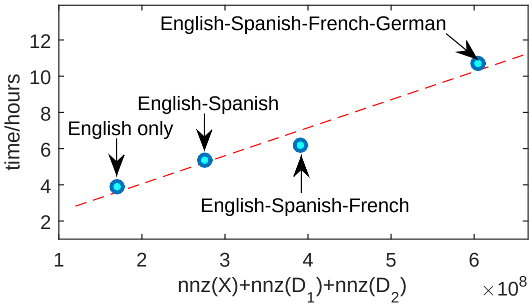

Figure 1: TI-LSA is linear in the number of nonzeros in the data matrices, and can easily scale to many languages.

Second, note that we have two dictionaries, one for the words and one for the contexts. These dictionaries correspond to the modes of the matrix; we have one dictionary matrix per mode, and we always multiply the dictionary along its corresponding mode. It would be easy to extend this method to a setting where the data is a 3-mode tensor instead of a matrix, e.g., if the data were (subject, verb, object) triples, or relation triples in some knowledge base. In these settings, the dictionaries used for each mode might be more different; in the subject-verb-object example, one of the dictionaries would only have verbs, while the other two would only have nouns, for instance. Standard tensor decompositions could be augmented with a translation-invariance term, similar to what we have done with matrices in this work.

6 Related Work

The most closely related work is that of Faruqui and Dyer (2014b), whose CCA-based method we have already mentioned; however, it is not obvious how CCA-based methods can be applied to more than two languages at a time. Our work is also similar to prior work on multilingual latent semantic analysis; Bader and Chew (2008) also include a translation dictionary when decomposing the **X** matrix, though their formulation uses a term-document matrix instead of a word-context

[1] This is in contrast to the results reported by Faruqui and Dyer, who by our evaluation also do not improve performance using multilingual data. To obtain word vectors from our decomposition, we used only the U component of the SVD; including the singular values, as Faruqui and Dyer did, gives worse performance. We confirmed this with the authors, and replicated their result for English-only LSA when using the singular values.

matrix, and the way they use the translation dictionary is quite different.

7 Conclusions

We have presented a new technique for generating word embeddings from multilingual corpora. This technique formalizes the notion of translation invariance into the objective function of the matrix decomposition and provides flexible and scalable means for obtaining word vectors where words that are translations of each other are close in the learned vector space. Through three separate evaluations, we showed that our technique gives superior performance on multilingual tasks, matches prior work on monolingual tasks, and scales linearly in the size of the input data. The code and data used in this paper are available at `https://sites.google.com/a/umn.edu/huang663/`.

Acknowledgements

Research was supported by the National Science Foundation Grant No. IIS-1247489, IIS-1247632, and a gift from Google. This work was also supported in part by a fellowship to Kejun Huang from the University of Minnesota Informatics Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

[Bader and Chew2008] Brett W Bader and Peter A Chew. 2008. Enhancing multilingual latent semantic analysis with term alignment information. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 49–56. Association for Computational Linguistics.

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

[Faruqui and Dyer2014a] Manaal Faruqui and Chris Dyer. 2014a. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, USA, June. Association for Computational Linguistics.

[Faruqui and Dyer2014b] Manaal Faruqui and Chris Dyer. 2014b. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, volume 2014.

[Firth1957] John R. Firth. 1957. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, pages 1–32.

[Ganitkevitch et al.2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

[Golub and Van Loan1996] Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations*. Johns Hopkins University Press, 3rd edition.

[Guo et al.2015] Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of ACL*.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR workshop*.