Accurate Cross-lingual Projection between Count-based Word Vectors by Exploiting Translatable Context Pairs

Shonosuke Ishiwatari[♣], **Nobuhiro Kaji**[♢♡], **Naoki Yoshinaga**[♢♡],
♣ Graduate School of Information Science and Technology, the University of Tokyo
♢ Institute of Industrial Science, the University of Tokyo
♡ National Institute of Information and Communications Technology, Japan
{ishiwatari, kaji, ynaga}@tkl.iis.u-tokyo.ac.jp

Masashi Toyoda[♢], **Masaru Kitsuregawa**[♢♠]
♢ Institute of Industrial Science, the University of Tokyo
♠ National Institute of Informatics, Japan
{toyoda, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

We propose a method that learns a cross-lingual projection of word representations from one language into another. Our method utilizes translatable context pairs as bonus terms of the objective function. In the experiments, our method outperformed existing methods in three language pairs, (English, Spanish), (Japanese, Chinese) and (English, Japanese), without using any additional supervisions.

1 Introduction

Vector-based representations of word meanings, hereafter *word vectors*, have been widely used in a variety of NLP applications including synonym detection (Baroni et al., 2014), paraphrase detection (Erk and Padó, 2008), and dialogue analysis (Kalchbrenner and Blunsom, 2013). The basic idea behind those representation methods is the distributional hypothesis (Harris, 1954; Firth, 1957) that similar words are likely to co-occur with similar context words.

A problem with the word vectors is that they are not meant for capturing the similarity between words in different languages, *i.e.*, translation pairs such as "*gato*" and "*cat*." The meaning representations of such word pairs are usually dissimilar, because the vast majority of the context words are from the same language as the target words (*e.g.*, Spanish for "*gato*" and English for "*cat*"). This prevents using word vectors in multi-lingual applications such as cross-lingual information retrieval and machine translation.

Several approaches have been made so far to address this problem (Fung, 1998; Klementiev et al., 2012; Mikolov et al., 2013b). In particular, Mikolov et al. (2013b) recently explored learning a linear transformation between word vectors of different languages from a small amount of training data, *i.e.*, a set of bilingual word pairs.

This study explores incorporating prior knowledge about the correspondence between dimensions of word vectors to learn more accurate transformation, when using count-based word vectors (Baroni et al., 2014). Since the dimensions of count-based word vectors are explicitly associated with context words, we can partially be aware of the cross-lingual correspondence between the dimensions of word vectors by diverting the training data. Also, word surface forms present noisy yet useful clues on the correspondence when targeting the language pairs that have exchanged their vocabulary (*e.g.*, "*cocktail*" in English and "*cóctel*" in Spanish). Although apparently useful, how to exploit such knowledge within the learning framework has not been addressed so far.

We evaluated the proposed method in three language pairs. Compared with baselines including a method that uses vectors learned by neural networks, our method gave better results.

2 Related Work

Neural networks (Mikolov et al., 2013a; Bengio et al., 2003) have recently gained much attention as a way of inducing word vectors. Although the scope of our study is currently limited to the count-based word vectors, our experiment demonstrated that the proposed method performs significantly better than strong baselines including neural networks. This suggests that count-based word vectors have a great advantage when learning a cross-lingual projection. As a future work, we are also

Proceedings of the 19th Conference on Computational Language Learning, pages 300–304,
Beijing, China, July 30-31, 2015.

interested in extending the method presented here to apply word vectors learned by neural networks.

There are also methods that directly inducing meaning representations shared by different languages (Klementiev et al., 2012; Lauly et al., 2014; Xiao and Guo, 2014; Hermann and Blunsom, 2014; Faruqui and Dyer, 2014; Gouws and Søgaard, 2015), rather than learning transformation between different languages (Fung, 1998; Mikolov et al., 2013b; Dinu and Baroni, 2014). However, the former approach is unable to handle words not appearing in the training data, unlike the latter approach.

3 Proposed Method

3.1 Learning cross-lingual projection

We begin by introducing the previous method of learning a linear transformation from word vectors in one language into another, which are hereafter referred to as source and target language.

Suppose we have a training data of n examples $\{(\boldsymbol{x}_1, \boldsymbol{z}_1), (\boldsymbol{x}_2, \boldsymbol{z}_2), \dots (\boldsymbol{x}_n, \boldsymbol{z}_n)\}$, where $\boldsymbol{x}_i$ is the count-based vector representation of a word in the source language (*e.g.*, "*gato*"), and $\boldsymbol{z}_i$ is the word vector of its translation in the target language (*e.g.*, "*cat*"). Then, we seek for a translation matrix, $\boldsymbol{W}$, such that $\boldsymbol{W}\boldsymbol{x}_i$ approximates $\boldsymbol{z}_i$, by solving the following optimization problem.

$$\boldsymbol{W}^\star = \operatorname*{argmin}_{\boldsymbol{W}} \sum_{i=1}^{n} \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{z}_i\|^2 + \frac{\lambda}{2}\|\boldsymbol{W}\|^2. \quad (1)$$

The second term is the L_2 regularizer. Although the regularization term does not appear in the original formalization (Mikolov et al., 2013b), we take this as a starting point of our investigation because the regularizer can prevent over-fitting and generally helps learn better models.

3.2 Exploiting translatable context pairs

Within the learning framework above, we propose exploiting the fact that dimensions of count-based word vectors are associated with context words, and some dimensions in the source language are translations of those in the target language.

For illustration purpose, suppose count-based word vectors of Spanish and English. The Spanish word vectors would have dimensions associated with context words such as "*amigo*," "*comer*," "*importante*," while the dimensions of the English word vectors are associated with "*eat*," "*run*," "*small*" and "*importance*," and so on. Since, for example, "*friend*" is a English translation of "*amigo*," the Spanish dimension associated with "*amigo*" is likely to be mapped to the English dimension associated with "*friend*." Such knowledge about the cross-lingual correspondence between dimensions is considered beneficial for learning accurate translation matrix.

We take two approaches to obtaining such correspondence. Firstly, since we have already assumed that a small amount of training data is available for training the translation matrix, it can also be used for finding the correspondence between dimensions (referred to as $\mathcal{D}_{train}$). Note that it is natural that some words in a language have many translations in another language. Thus, for example, $\mathcal{D}_{train}$ may include ("*amigo*", "*friend*"), ("*amigo*", "*fan*") and ("*amigo*", "*supporter*").

Secondly, since languages have evolved over the years while often deriving or borrowing words (or concepts) from those in other languages, those words have similar or even the same spelling. We take advantage of this to find the correspondence between dimensions. We specifically define function $\textsc{dist}(r, s)$ that measures the surface-level similarity, and regard all context word pairs (r, s) having smaller distance than a threshold[1] as translatable ones (referred to as $\mathcal{D}_{sim}$).

$$\textsc{dist}(r, s) = \frac{Levenshtein(r, s)}{\min(len(r), len(s))}$$

where function $Levenshtein(r, s)$ represents the Levenshtein distance between the two words, and $len(r)$ represents the length of the word.

3.3 New objective function

We incorporate the knowledge about the correspondence between the dimensions into the learning framework. Since the correspondence obtained by the methods presented above can be noisy, we want to treat it as a soft constraint. This consideration leads us to develop the following new objective function:

$$\boldsymbol{W}^\star = \operatorname*{argmin}_{\boldsymbol{W}} \sum_{i=1}^{n} \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{z}_i\|^2 + \frac{\lambda}{2}\|\boldsymbol{W}\|^2 - \beta_{train} \sum_{(j,k)\in\mathcal{D}_{train}} w_{jk} - \beta_{sim} \sum_{(j,k)\in\mathcal{D}_{sim}} w_{jk}.$$

The third and fourth terms are newly added to guide the learning process to strengthen w_{jk} when

[1] The threshold was fixed to 0.5.

k-th dimension in the source language corresponds to j-th dimension in the target language. $\mathcal{D}_{train}$ and $\mathcal{D}_{sim}$ are sets of dimension pairs found by the two methods. β_{train} and β_{sim} are parameters representing the strength of the new terms, and are tuned on held-out development data.

3.4 Optimization

We use Pegasos algorithm (Shalev-Shwartz et al., 2011), an instance of the stochastic gradient descent (Bottou, 2004), to optimize the new objective. Given τ-th learning sample $(\boldsymbol{x}_\tau, \boldsymbol{z}_\tau)$, we update translation matrix $\boldsymbol{W}$ as follows:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta_\tau \nabla E_\tau(\boldsymbol{W})$$

where η_τ represents the learning rate and is set to $\eta_\tau = \frac{1}{\lambda \tau}$, and $\nabla E_\tau(\boldsymbol{W})$ is the gradient which is calculated from τ-th sample $(\boldsymbol{x}_\tau, \boldsymbol{z}_\tau)$:

$$2(\boldsymbol{W}\boldsymbol{x}_\tau - \boldsymbol{z}_\tau)\boldsymbol{x}_\tau^{\mathrm{T}} - \beta_{train}\boldsymbol{A} - \beta_{sim}\boldsymbol{B} + \lambda \boldsymbol{W}.$$

$\boldsymbol{A}$ and $\boldsymbol{B}$ are gradients corresponding to the two new terms. $\boldsymbol{A}$ is a matrix in which $a_{jk} = 1$ if $(j, k) \in \mathcal{D}_{train}$ otherwise 0. $\boldsymbol{B}$ is defined similarly.

4 Experiments

We evaluate our method on translation among word vectors in four languages: English (En), Spanish (Es), Japanese (Jp) and Chinese (Cn). We have chosen three language pairs: (En, Es), (Jp, Cn) and (En, Jp), for the translation, so that we can examine the impact of each type of translatable context pairs integrated into the learning objective.

4.1 Setup

First, we prepared source text in the four languages from Wikipedia[2] dumps following (Baroni et al., 2014). We extracted plain text from the XML dumps by using wp2txt.[3] Since words are concatenated in Japanese and Chinese, we used MeCab[4] and Stanford Word Segmenter[5] to tokenize the text. Since inflection occurs in English, Spanish, and Japanese, we used Stanford POS tagger,[6] Pattern,[7] and MeCab to lemmatize the text.

[2]http://dumps.wikimedia.org/
[3]https://github.com/yohasebe/wp2txt/
[4]http://taku910.github.io/mecab/
[5]http://nlp.stanford.edu/software/segmenter.shtml
[6]http://nlp.stanford.edu/software/tagger.shtml
[7]http://www.clips.ua.ac.be/pages/pattern

Next, we induced count-based word vectors from the obtained text. We considered context windows of five words to both sides of the target word. The function words are then excluded from the extracted context words. Since the count vectors are very high-dimensional and sparse, we selected top-10k frequent words as contexts words (in other words, the number of dimensions of the word vectors). We converted the counts into positive point-wise mutual information (Church and Hanks, 1990) and normalized the resulting vectors to remove the bias that is introduced by the difference of the word frequency.

Then, we compiled a seed bilingual dictionary (a set of bilingual word pairs) for each language pair that is used to learn and evaluate the translation matrix. We utilized cross-lingual synsets in the Open Multilingual Wordnet[8] to obtain bilingual pairs.

Since our method aims to be used in expanding bilingual dictionaries, we designed datasets assuming such a situation. Considering that more frequent words are likely to be registered in a dictionary, we sorted words in the source language by frequency and used the top-11k words and their translations in the target language as a training/development data, and used the subsequent 1k words and their translations as a test data.

We have compared our method with the following three methods:

Baseline learns a translation matrix using Eq. 1 for the same count-based word vectors as the proposed method. Comparison between the proposed method and this method reveals the impact of incorporating the cross-lingual correspondences between dimensions.

CBOW learns a translation matrix using Eq. 1 for word vectors learned by a neural network (specifically, continuous bag-of-words (CBOW)) (Mikolov et al., 2013b). Comparison between this method and the above baseline reveals the impact of the vector representation. Note that the CBOW-based word vectors take rare context words as well as the top-10k frequent words into account. We used word2vec[9] to obtain the vectors for each language.[10] Since Mikolov et al. (2013b)

[8]http://compling.hss.ntu.edu.sg/omw/
[9]https://code.google.com/p/word2vec/
[10]The threshold of sub-sampling of words was set to 1e-3 to reduce the effect of very frequent words, *e.g.*, "*a*" or "*the*."

Table 1: Experimental results: the accuracy of the translation.

Testset	Baseline		CBOW		Direct Mapping		Proposed$_{\text{w/o surface}}$		Proposed	
	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5
Es → En	0.1%	0.5%	7.5%	22.0%	45.7%	61.1%	46.6%	62.4%	**54.7%**	**67.6%**
Es ← En	0.1%	0.6%	7.1%	18.9%	11.9%	26.1%	28.7%	45.7%	**31.3%**	**49.6%**
Jp → Cn	0.6%	1.6%	5.4%	13.8%	9.3%	22.2%	11.1%	26.2%	**15.5%**	**34.0%**
Jp ← Cn	0.3%	1.2%	2.9%	11.3%	11.6%	26.8%	7.8%	21.6%	**13.1%**	**27.9%**
En → Jp	0.3%	1.1%	4.9%	13.3%	5.4%	13.9%	18.5%	36.4%	**19.3%**	**37.1%**
En ← Jp	0.2%	1.0%	6.5%	19.1%	22.3%	37.4%	32.3%	51.0%	**32.5%**	**51.9%**

reported the accurate translation can be obtained when the vectors in the source language is 2-4x larger than that in the target language, we prepared m-dimensional ($m = 100, 200, 300$) vectors for the target language and n-dimensional ($n = 2m, 3m, 4m$) vectors for the source language, and optimized their combinations on the development data.

Direct Mapping exploits the training data to map each dimension in a word vector in the source language to the corresponding dimension in a word vector in the target language, referring to the bilingual pairs in the training data (Fung, 1998). To deal with words that have more than one translation, we weighted each translation by a reciprocal rank of its frequency among the translations in the target language, as in (Prochasson et al., 2009).

Note that all methods, including the proposed methods, use the same amount of supervision (training data) and thereby they are completely comparable with each other.

Evaluation procedure For each word vector in the source language, we translate it into the target language and evaluate the quality of the translation as in (Mikolov et al., 2013b): i) measure the cosine similarity between the resulting word vector and all the vectors in the test data (in the target language), ii) next choose the top-n ($n = 1, 5$) word vectors that have the highest similarity against the resulting vector, and iii) then examine whether the chosen vectors include the correct one.

4.2 Results

Table 1 shows results of the translation between word vectors in each language pair. **Proposed** significantly improved the translation quality against **Baseline**, and performed the best among all of the methods. Although the use of CBOW-based word vectors (**CBOW**) has improved the translation quality against **Baseline**, the performance

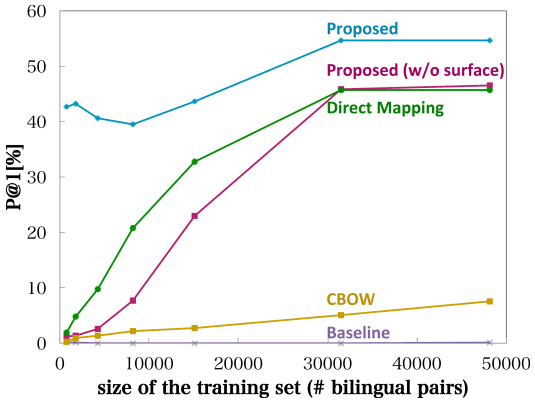

Figure 1: The impact of the size of training data (Es → En).

gain is smaller than that obtained by our new objective. **Proposed$_{\text{w/o surface}}$** uses only the training data to find translatable context pairs by setting $\beta_{sim} = 0$. Thus, its advantage over **Direct Mapping** confirms the importance of learning a translation matrix. In addition, the greater advantage of **Proposed** over **Proposed$_{\text{w/o surface}}$** in the translation between (En, Es) or (Jp, Cn) conforms to our expectation that surface-level similarity is more useful for translation between the language pairs which have often exchanged their vocabulary.

Figure 1 shows P@1 (Es → En) plotted against the size of training data. Remember that the training data is not only used to learn a translation matrix in the methods other than **Direct Mapping** but also is used to map dimensions in **Direct Mapping** and the proposed methods. **Proposed** performs the best among all methods regardless the size of training data. Comparison between **Direct Mapping** and **Proposed$_{\text{w/o surface}}$** reveals that learning a translation matrix is not always effective when the size of the training data is small, since it may be suffered from over-fitting (the size of the translation matrix is too large for the size of training data). We can see that surface-level similarity is beneficial especially when the size of training data is small.

5 Conclusion

We have proposed the use of prior knowledge in accurately translating word vectors. We have specifically exploited two types of translatable context pairs, which are taken from the training data and guessed by surface-level similarity, to design a new objective function in learning the translation matrix. Experimental results confirmed that our method significantly improved the translation among word vectors in four languages, and the advantage was greater than that obtained by the use of a word vector learned by a neural network.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 25280111.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, pages 238–247.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Léon Bottou. 2004. Stochastic learning. In *Advanced lectures on machine learning*, pages 146–168. Springer.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Georgiana Dinu and Marco Baroni. 2014. How to make words with vectors: Phrase generation in distributional semantics. In *Proceedings of ACL*, pages 624–633.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*, pages 897–906.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, pages 462–471.

John R. Firth. 1957. A synopsis of linguistic theory. *Studies in Linguistic Analysis*, pages 1–32.

Pascale Fung. 1998. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In *Machine Translation and the Information Soup*, pages 1–17. Springer.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of NAACL-HLT*, pages 1386–1390.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10:146–162.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of ACL*, pages 58–68.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of ACL Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING*, pages 1459–1474.

Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Advances in NIPS*, pages 1853–1861.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

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

Emmanuel Prochasson, Emmanuel Morin, and Kyo Kageura. 2009. Anchor points for bilingual lexicon extraction from small comparable corpora. In *Proceedings of MT Summit XII*, pages 284–291.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. 2011. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical programming*, 127(1):3–30.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of CoNLL*, pages 119–129.