Injecting Word Embeddings with Another Language's Resource : An Application of Bilingual Embeddings

by

Prakhar Pandey, Vikram Pudi, Manish Shrivastava

in

International Joint Conference on Natural Language Processing
(*IJCNLP-2017*)

Taipei, Taiwan

Report No: IIIT/TR/2017/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2017

Injecting Word Embeddings with Another Language's Resource : An Application of Bilingual Embeddings

Prakhar Pandey **Vikram Pudi** **Manish Shrivastava**
International Institute of Information Technology
Hyderabad, Telangana, India
prakhar.pandey@research.iiit.ac.in
{vikram,manish.shrivastava}@.iiit.ac.in

Abstract

Word embeddings learned from text corpus can be improved by injecting knowledge from external resources, while at the same time also specializing them for similarity or relatedness. These knowledge resources (like WordNet, Paraphrase Database) may not exist for all languages. In this work we introduce a method to inject word embeddings of a language with knowledge resource of another language by leveraging bilingual embeddings. First we improve word embeddings of German, Italian, French and Spanish using resources of English and test them on variety of word similarity tasks. Then we demonstrate the utility of our method by creating improved embeddings for Urdu and Telugu languages using Hindi WordNet, beating the previously established baseline for Urdu.

1 Introduction

Recently fast and scalable methods to generate dense vector space models have become very popular following the works of (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014). These methods take large amounts of text corpus to generate real valued vector representation for words (word embeddings) which carry many semantic properties.

Mikolov et al. (2013b) extended this model to two languages by introducing bilingual embeddings where word embeddings for two languages are simultaneously represented in the same vector space. The model is trained such that word embeddings capture not only semantic information of monolingual words, but also semantic relationships across different languages. A number of different methods have since been proposed to construct bilingual embeddings (Zou et al., 2013; Vulic and Moens, 2015; Coulmance et al., 2016).

A disadvantage of learning word embeddings only from text corpus is that valuable knowledge contained in knowledge resources like WordNet (Miller, 1995) is not used. Numerous methods have been proposed to incorporate knowledge from external resources into word embeddings for their refinement (Xu et al., 2014; Bian et al., 2014; Mrksic et al., 2016). (Faruqui et al., 2015) introduced *retrofitting* as a light graph based technique that improves learned word embeddings.

In this work we introduce a method to improve word embeddings of one language (target language) using knowledge resources from some other similar language (source language). To accomplish this, we represent both languages in the same vector space (bilingual embeddings) and obtain translations of source language's resources. Then we use these translations to improve the embeddings of the target language by using *retrofitting*, leveraging the information contained in bilingual space to adjust retrofitting process and handle noise. We also show why a dictionary based translation would be ineffective for this problem and how to handle situations where vocabulary of target embeddings is too big or too small compared to size of resource.

(Kiela et al., 2015) demonstrated the advantage of specializing word embeddings for either similarity or relatedness, which we also incorporate. Our method is also independent of the way bilingual embeddings were obtained. An added advantage of using bilingual embeddings is that they are better than monolingual counterparts due to incorporating multilingual evidence (Faruqui and Dyer, 2014; Mrkšić et al., 2017).

2 Background

2.1 Bilingual Embeddings

Various methods have been proposed to generate bilingual embeddings. One class of methods learn mappings to transform words from one monolingual model to another, using some form of dictionary (Mikolov et al., 2013b; Faruqui and Dyer, 2014). Another class of methods jointly optimize monolingual and cross-lingual objectives using word aligned parallel corpus (Klementiev et al., 2012; Zou et al., 2013) or sentence aligned parallel corpus (Chandar A P et al., 2014; Hermann and Blunsom, 2014). Also there are other methods which use monolingual data and a smaller set of sentence aligned parallel corpus (Coulmance et al., 2016) and those which use non-parallel document aligned data (Vulic and Moens, 2015).

We experiment with *translation invariant* bilingual embeddings by (Gardner et al., 2015). We also experiment with method proposed by (Artetxe et al., 2016) where they learn a linear transform between two monolingual embeddings with *monolingual invariance preserved*. They use a small bilingual dictionary to accomplish this task. These methods are useful in our situation because they preserve the quality of original monolingual embeddings and do not require parallel text (beneficial in case of Indian languages).

2.2 Retrofitting

Retrofitting was introduced by (Faruqui et al., 2015) as a light graph based procedure for enriching word embeddings with semantic lexicons. The method operates *post processing* i.e it can be applied to word embeddings obtained from any standard technique such as Word2vec, Glove etc. The method encourages improved vectors to be similar to the vectors of *similar words* as well as similar to the original vectors. This similarity relation among words (such as synonymy, hypernymy, paraphrase) is derived from a knowledge resource such as PPDB, WordNet etc. Retrofitting works as follows:

Let matrix Q contain the word embeddings to be improved. Let $V = \{w_1, w_2 ... w_n\}$ be the vocabulary which is equal to number of rows in Q and d be the dimension of word vectors which is equal to number of columns. Also let Ω be the ontology that contains the intra word relations that must be injected into the embeddings. The objective of retrofitting is to find a matrix $\hat{Q}$ such that the new word vectors are close to their original vectors as well as vectors of related words. The function to be minimized to accomplish this objective is:

$$\Phi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

The iterative update equation is:

$$q_i = \frac{\sum_{j:(i,j) \in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(i,j) \in E} \beta_{ij} + \alpha_i}$$

α and β are the parameters used to control the process. We discuss in Section 3.2 how we set them to adapt the process to bilingual settings.

2.3 Dictionary based approach

Before discussing our method, we would like to point that using a dictionary for translating the lexical resource and then retrofitting with this translated resource is not feasible. Firstly obtaining good quality dictionaries is a difficult and costly process[1]. Secondly it is not necessary that one would obtain translations that are within the vocabulary of the embeddings to be improved. To demonstrate this, we obtain translations for embeddings of 3 languages[2] and show the results in Table 1. In all cases the number of translations that are also present in the embedding's vocabulary are too small.

Language	Vocab	Matches
German	43,527	9,950
Italian	73,427	24,716
Spanish	41,779	16,547

Table 1: Using a dictionary based approach

3 Approach

Let S, T and R be the vocabularies of source, target and resource respectively. Size of R is always fixed while size of S and T depends on embeddings. The relation between S, T and R is shown in Figure 1. Sets S and R have one to one mapping which in not necessarily onto, while T and S have many to one mapping. Consider the ideal case where every word in R is also in S and every word in S has the exact translation from T as its nearest neighbour in the bilingual space. Then the

[1] eg. Google and Bing Translate APIs have become paid.

[2] using Yandex Translate API, it took around 3 days

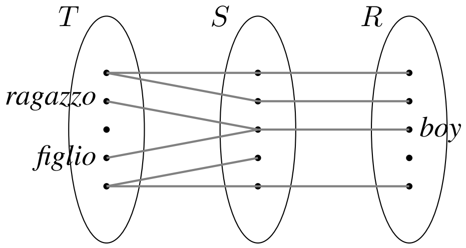

Figure 1: Relationships between Source, Target and Resource Vocabularies.

simple approach for translation would be assigning every $s_i \in S$ its nearest neighbour $t_i \in T$ as the translation.

First problem is that in practical settings these conditions are almost never satisfied. Secondly the sizes of S, T and R can be very different. Suppose the size of S, T is large compared to R or the size of T is large but size of S is comparatively smaller. In both cases size of translated resource will be too small to make impact. Thirdly words common to both R and S will be lesser than the total words in R. So the size of R accessible to T using S will be even lesser. A mechanism is therefore required to control the size of translated resource and filter incorrect translations. We accomplish this as follows:

3.1 Translating knowledge resource

For translation we adapt a dual approach that allows control over the size of the translated list. We iterate through T (not S) looking for translations in S. A translation is accepted or rejected based on whether the cosine similarity between words is above the threshold η. This method stems from the fact that mapping between T and S is many to one. So the Italian word *ragazzo* is translated to *boy*, but we can also translate (and subsequently retrofit) *figlio* to *boy* (Figure 1) in order to get a larger translated resource list with some loss in quality of list. Thus η gives us direct control over the size of translated resource list. Then to translate the list of related words, we translate normally (i.e from S to T). Algorithm 1 summarizes this process.

3.2 Retrofitting to translated resource

We modify the retrofitting procedure to accommodate noisy translations as follows:

As discussed earlier, retrofitting process controls the relative shift of vectors by two parameters α and β, where α controls the movement towards original vector and β controls movement towards vectors to be fitted with. (Faruqui et al., 2015) set α as 1 and β as $\frac{1}{\gamma}$ where γ is the number of vectors to be fitted with. Thus they give equal weights to each vector.

Cosine similarity between a word and its translation is a good measure of the confidence in its translation. We use it to set β such that different vectors get different weights. A word for which we have higher confidence in translation (i.e higher cosine similarity) is given more weight when retrofitting. Therefore w_i being the weights, α, β are set as :

$$\alpha = \sum_{i=1}^{\gamma} w_i, \quad \beta_i = w_i$$

Further reduction in weights of noisy words can be done by taking powers of cosine similarity. An example in Table 2 shows weights of similar words for Italian word *professore* derived by taking powers of cosine similarity (we refer to this power as *filter* parameter).

Words	**Similarity**	**Weights**
educatore	0.955	0.796
harvard	0.853	0.452
insegnando	0.980	0.903
insegnata	0.990	0.951

Table 2: Taking power of weights reduces weights of noisy words (here *harvard*). Here $filter = 5$.

Algorithm 1 Translating Knowledge Resource

```
Input : Source ($S$), Target ($T$), Resource ($R$), $\eta$
Output : Translated Knowledge Resource $R^*$
  $R^* = []$
  for $t$ in $T$ do
    $t_s \leftarrow NearestNeighbour(S)$
    if $similarity(t, t_s) > \eta$ then
      $lexicons \leftarrow S[t_s]$
      for $l$ in $lexicons$ do
        $l_t \leftarrow NearestNeighbour(T)$
        $weight \leftarrow similarity(l, l_t)$
        $R^*[t].add(l_t, weight)$
      end for
    else
      $continue$
    end if
  end for
```

Language	Vocab	TRL	Tasks	Original	Half Enriched	Full Enriched
German	43,527	18,802 37,408	MC30 RG65 WS353 (sim) Simlex999	0.631 0.503 0.600 0.333	0.643 0.531 0.631 0.356	0.662 0.600 0.635 0.373
Italian	73,427	22,022 44,309	WS353 (sim) Simlex999	0.595 0.247	0.640 0.283	0.652 0.313
Spanish	41,779	17,434 35,034	MC30 RG65	0.312 0.608	0.286 0.615	0.412 0.634
French	40,523	16,203 32,602	RG65	0.547	0.582	0.673

Table 3: Retrofitting Translation Invariant Bilingual Embeddings for German, Italian, Spanish and French using English Paraphrase Database. (TRL stands for Translated Resource Length)

4 Datasets and Benchmarks

For English as source language, we use the Paraphrase Database (Ganitkevitch et al., 2013) to specialize embeddings for similarity as it gives the best results (compared to other sources like WordNet). To specialize embeddings for relatedness, we use University of South Florida Free Association Norms (Nelson et al., 2004) as indicated by (Kiela et al., 2015). For Hindi as source language, we use Hindi WordNet (Bhattacharyya et al., 2010). Whenever the size of resource is big enough, we first inject word embeddings with half of the dataset (random selection) followed by full length dataset to demonstrate the sequential gain in performance.

Multilingual WS353 and SimLex999 datasets are by (Leviant and Reichart, 2015). We also use German RG65 (Gurevych, 2005), French RG65 (Joubarne and Inkpen, 2011) and Spanish MC30, RG65 (Hassan and Mihalcea, 2009; Camacho-Collados et al., 2015). For Indian languages we use datasets provided by (Akhtar et al., 2017).

5 Results

In this section we present the experimental results of our method.[3] Before discussing the results we explain how different parameters are chosen. We do 10 iterations of retrofitting process for all our experiments because 10 iterations are enough for convergence (Faruqui et al., 2015) and also using the same value for all experiments avoids overfitting. The value of $filter$ parameter is set as 2 because we believe the embeddings that we use are well trained and low in noise. This value can be increased further if word embeddings being used are very noisy but in most cases a value of 2 is enough. η value, as explained in previous sections is set such that the translated resource obtained is of sufficient length. If more lexicons in translated resource are required, relax η and vice-versa.

5.1 European Languages

Table 3 shows the result of retrofitting *translation invariant bilingual* embeddings of four European languages for similarity using English Paraphrase Database. For each language we set η as 0.70 and $filter$ as 2. The embeddings are evaluated specifically on datasets measuring similarity. All embeddings are 40 dimensional. To show that our method is effective, the embeddings are first fitted with only half of the database followed by fitting with full length database. Table 3 also contains information about the size of vocabulary and translated resource. One can compare the size of translated resource that we get using our method to the dictionary based approach.

Table 4 shows the results of specializing word embeddings for relatedness using the USF Association Norms and evaluation on WS353 relatedness task. We test only with German and Italian as only these languages had datasets to test for relatedness.

Language	Original	Fitted
German	0.461	0.520
Italian	0.460	0.523

Table 4: Specializing for relatedness

[3] The implementation of our method is available at https://github.com/prakhar987/InjectingBilingual

We also experiment with embeddings of large dimensions (300) and large vocabulary size (200,000) for English and Italian bilingual embeddings obtained by method described by (Artetxe et al., 2016). Table 5 shows the improvements attained for similarity task for Italian with 64,434 words in the translated resource, $\eta = 0.35$ and $filter = 2$ (notice η is much smaller since we want translated resource size to be comparable to size of vocabulary).

Task	Original	Fitted
WS353	0.648	0.680
SimLex999	0.371	0.405

Table 5: Improving large embeddings

5.2 Indian Languages

To demonstrate the utility of our method, we experiment with Indian languages, taking Hindi as the source language (which has Hindi WordNet). For target language, we take one language (Urdu) which is very similar to Hindi (belongs to same family) and one language (Telugu) which is very different from Hindi (descendants from same family). The vocabulary size of Urdu and Telugu were 129,863 and 174,008 respectively. The results are shown in Table 6. Here again we fit with half length followed by full length of Hindi WordNet. As expected, we get much higher gains for Urdu compared to Telugu.[4]

Language	Original	Half Fitted	Full Fitted
Telugu	0.427	0.436	0.440
Urdu	0.541	0.589	0.612

Table 6: Retrofitting Indian languages

6 Conclusion

In this work we introduced a method to improve word embeddings of a language using resources from another similar language. We accomplished this by translating the resource using bilingual embeddings and modifying retrofitting while handling noise. Using our method, we also created new benchmark on Urdu word similarity dataset.

[4] enriched embeddings and evaluation scripts can be downloaded from `https://goo.gl/tN6p3w`

References

Syed Sarfaraz Akhtar, Arihant Gupta, Avijit Vajpayee, Arjit Srivastava, and Manish Shrivastava. 2017. Word similarity datasets for indian languages: Annotation and baseline systems. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 91–94. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 2289–2294.

Pushpak Bhattacharyya, Prabhakar Pande, and Laxmi Lupu. 2010. Hindi wordnet. Language Resources and Evaluation (LRE).

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8724*, pages 132–148.

Jos Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A framework for the construction of monolingual and cross-lingualword similarity datasets. In *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, volume 2, pages 1–7.

Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Proceedings of Neural Information Processing Systems*, pages 1853–1861.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2016. Transgram, fast cross-lingual word-embeddings. *CoRR*, abs/1601.02502.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*.

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

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764.

Matt Gardner, Kejun Huang, Evangelos Papalexakis, Xiao Fu, Partha Talukdar, Christos Faloutsos, Nicholas Sidiropoulos, and Tom Mitchell. 2015. Translation invariant word embeddings. In *Proceedings of EMNLP*.

Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. *IJCNLP 2005*, pages 767–778.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, pages 1192–1201.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *CoRR*, abs/1404.4641.

Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the google n-gram corpus and second- order co-occurrence measures. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, pages 216–221.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of EMNLP*.

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

Ira Leviant and Roi Reichart. 2015. Judgment language matters: Multilingual vector space models for judgment language aware lexical semantics. *CoRR*, abs/1508.00106.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, pages 39–41.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. *In Proceedings of NAACL*, pages 142–148.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*.

Douglas L. Nelson, Cathy L. McEvoy, and Thomas A. Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 12, pages 1532–1543.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Volume 2: Short Papers*, pages 719–725.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1219–1228.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*, pages 1393–1398.