Exploring the effect of semantic similarity for Phrase-based Machine Translation

Kunal Sachdeva, Dipti Misra Sharma
Language Technologies Research Centre, IIIT Hyderabad
kunal.sachdeva@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

The paper investigates the use of semantic similarity scores as feature in the phrase based machine translation system. We propose the use of partial least square regression to learn the bilingual word embedding using compositional distributional semantics. The model outperforms the baseline system which is shown by an increase in BLEU score. We also show the effect of varying the vector dimension and context window for two different approaches of learning word vectors.

1 Introduction

The current state of the art Statistical Machine Translation (SMT) systems (Koehn et al., 2003) do not account for semantic information or semantic relatedness between the corresponding phrases while decoding the n-best list. The phrase pair alignments extracted from the parallel corpora offers further limitation of capturing contextual and linguistic information. Since the efficiency of statistical system depends on the quality of parallel corpora, low resourced language pair fails to meet the desired standards of translation.

Word representation is being widely used in many Natural Language Processing (NLP) applications like information retrieval, machine translation and paraphrasing. The word representation computed from continuous monolingual text provide useful information about the relationship between different words. Distributional semantics offers a notion of capturing semantic similarity between words occurring in similar context, where similar meaning words are grouped closely in a high dimension word space model. Each word is associated with an n-dimensional vector which represents its position in a vector space model and similar words are at small distance in comparison to relatively opposite meaning words.

The recent work in word vectors have shown to capture the linguistic relations and regularities. The relation between words can be expressed as a simple mathematical relation between their corresponding word vectors. The recent paper by Mikolov (Mikolov et al., 2013c) have shown through a word analogy task that the vec (”man”) - vec (”woman”) + vec(”king”) should be close to vec(”queen”). Capturing of these relations along with word composition have shown significant improvements in various NLP and information retrieval tasks.

In this paper, we present our ideas of capturing the semantic similarity between phrase pairs in context of SMT and use the scores as features while decoding n-best list. We make use of word representations computed from two different methods: word2Vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) and show the effect of varying the context window and vector dimension for Hindi-English language pair. We use partial least squares (PLS) regression to learn the bilingual word embeddings using a bilingual dictionary, which is most readily available resource for any language pair. In this work we are not optimizing over the vector dimension and context window, but provide insights (through experiments) on how these two parameters effect the similarity tasks.

The rest of the paper is organized as follows. We first present the related work in vector space models and their utilization in machine translation domain (section 2). Section 3 describes the two methods we have adopted for computing word embeddings. The basic SMT setup, formulating transformation model and phrase similarity scores are described in section 4. In section 5 we present our results and conclude the paper in section 6 with some future directions.

Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), pages 41–47,
Beijing, China, July 26-31, 2015.

2 Related Work

The current research community has shown special interest towards vector space models by organizing various dedicated workshops in top rated conferences. Word representations have been used in many NLP applications like information extraction (Paşca et al., 2006; Manning et al., 2008), sentiment prediction (Socher et al., 2011) and paraphrase detection (Huang, 2011).

In the past various methodologies have been suggested to learn bilingual word embeddings for various natural language related tasks. (Mikolov et al., 2013b) and (Zou et al., 2013) have shown significant improvements by using bilingual word embeddings in context of machine translation experiments. The former applies linear transformation to bilingual dictionary while the latter uses word alignments knowledge. Zhang (2014) proposed an auto-encoder based approach to learn phrase embeddings from the word vectors and showed improvements by using semantic similarity score in MT experiments. The phrase vector is generated by recursively combining the two children vector into a same dimensional parent vector using the method suggested by (Socher et al., 2011).

The work of (Gao et al., 2013) proposes a method for learning the semantic representation of phrase using features (multi-layer neural network) which is then used to compute the distance between them in a low dimensional space. The learning of weights in the neural network is guided by the BLEU score (ultimate goal to improve the quality of translation through increase in BLEU score) which makes it sensitive towards the score. Wu (2014) proposed an approach of using supervised model of learning context-sensitive bilingual embedding where the aligned phrase pairs are marked as true labels.

Since these defined methods depends heavily on the quality of word vectors, a number of approaches have been suggested in past to learn word representations from monolingual corpus: word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) and (Huang et al., 2012).

In this work, we extend the phrase similarity work by using the regression approach to learn the bilingual word embeddings. We employ vector composition approach to compute the phrase vector, where we add vectors of each constituent word to achieve the phrase vector. We also present the comparison of using different word embedding models along with varying context window and vector dimension which has not been shown (in detail) in any of the previous works. As pointed by (Mikolov et al., 2013b) linear transformation works well for language pairs which are closely related, however in this work we experiment with PLS regression which also establishes a linear relationship between words but is much more efficient than the simple least squares regression (explained in 4.2).

3 Learning word representation

We have used a part of WMT'14[1] monolingual data and news crawled monolingual data to learn word representations for English and Hindi respectively. We added the ILCI bilingual corpus (Jha, 2010) of English and Hindi to the monolingual data. The corpus statistics (after cleaning) are provided in table 1. The vocabulary refers to the words in embeddings with a minimum frequency of five within the corpus.

Language	# of Words	Vocabulary
English	250M	274K
Hindi	80M	184K

Table 1: Monolingual corpus statistics

3.1 word2Vec

The word2Vec model proposed by (Mikolov et al., 2013a) computes vectors by skip-gram and continuous bag of words (CBOW) model. These models use a single layer neural networks and are computationally much more efficient than any previously proposed model. The CBOW architecture of model predicts the current word based on the context whereas the skip-gram model predicts the neighboring words depending on the current word. Experiments have shown CBOW architecture to perform better on the syntactic task and skip-gram based architecture on the semantic tasks.

We have used the skip-gram architecture of word2Vec in our experiments as it has been shown to perform better for semantic related tasks.

3.2 GloVe

The Global Vector model of learning word representation was proposed by (Pennington et al., 2014) which computes the word vectors from a

[1] http://www.statmt.org/wmt14/translation-task.html

global word-word co-occurrence matrix. The relationship between words is extracted by using the ratio of co-occurrence probability with various probe words, which distinguishes between the relevant and irrelevant words. The co-occurrence probability of word *'i'* to that of word *'j'* is studied on the basis of a probe word *'k'* which is computed on the basis of a ratio P_{ik}/P_{jk}. The ratio is expected to be higher if word *'k'* is more related to word *'i* and low if it is related to word *'j'*. The author shows significant improvement over the word2Vec model on various NLP tasks (word similarity, word analogy and named entities recognition).

For training both the models we have altered the vector size and the context window, while all other parameters are set to default.

4 Experiments

4.1 Baseline MT System

We have used the ILCI corpora (Jha, 2010) which contains 50000 Hindi-English parallel sentences (49300 after cleaning) from health and tourism domains. The corpus is randomly split (equal variation of sentence length) into training (48300 sentences), development (500 sentences) and testing (500 sentences).

Division	# of sentences
Training	48300
Development	500
Testing	500

Table 2: MT system corpus statistics

We trained two Phrase based (Koehn et al., 2003) MT systems (Hindi - English and English - Hindi) using the Moses toolkit (Koehn et al., 2007) with phrase-alignments (maximum phrase length restricted to 4) extracted from GIZA++ (Och and Ney, 2000). We have used the SRILM (Stolcke and others, 2002) with Kneser-Ney smoothing (Kneser and Ney, 1995) for training a language model of order five and MERT (Och, 2003) for tuning the model with development data. We achieve a BLEU (Papineni et al., 2002) score of 19.89 and 22.82 on English-Hindi and Hindi-English translation systems respectively. These translation scores serves as our baseline for further experiments.

4.2 Partial Least Square (PLS) Regression

We generate the word embeddings of both Hindi and English using monolingual corpus using two previously mentioned methods (section 3). Since both the word embeddings are in different space (computed independently), there is a need to map the source vector space to target vector space or vice versa.

We employ the PLS (Abdi, 2003) regression to learn the transformation matrices. The observable variables (X) are the word embeddings of one language, while the predictable variables (Y) are the word embeddings of the other language. The observable and the predictable are $n \times d$ matrices, where 'n' is the number of words used (explained in subsection 4.3) and 'd' is the word embedding dimension. Our task is to compute a transformation matrix of $d \times d$ dimension which will be used to transform any given language word vector to its corresponding other language vector.

The PLS[2] regression algorithm works by projecting both X and Y matrices to a new space, and decomposes them into a set of orthogonal factors. The observables are first decomposed as $T = XW$ where 'T' and 'W' are the factor score matrix and weight matrix respectively. The predictable 'Y' is then estimated as $Y = TQ+E$ where 'Q' and 'E' are regression coefficient matrix and error term. We have the final regression model as $Y = XB+E$ where $B = WQ$ acts as our transformation matrix.

Dimension	word2Vec		GloVe	
	CW 5	CW 7	CW 5	CW 7
50	0.53	0.51	0.48	0.49
100	0.47	0.49	0.43	0.44
150	0.44	0.47	0.41	0.42
200	0.42	0.45	0.38	0.41
250	0.41	0.43	0.38	0.39
300	0.40	0.41	0.37	0.39
400	0.40	0.38	0.35	0.36
500	0.38	0.37	0.34	0.36

Table 3: Average word cosine similarity scores on test set. Context Window (CW)

4.3 Learning Transformation matrix

We employ PLS regression to learn bilingual word embeddings using a English-Hindi bilingual dictionary [3]. We have used 15000 words for train-

[2]http://www.statsoft.com/Textbook/Partial-Least-Squares
[3]http://www.shabdkosh.com/

ing the regression model and another set of 1500 words for testing purpose. The bilingual pair of training words are selected based on the frequency of those words occurring in a large plain text which consist of 10000 words from high frequency and 2500 words each of low and medium frequencies.

The observable variable and the predictable variables in the PLS regression are the word vectors of each word pair from their respective language word embedding models. We finally achieve two transformation models which transforms source to target vector space and target to source vector space. We have presented average similarity score on the test set in table 3 after transforming English words to Hindi word space.

4.4 Decoding with semantic similarity score

In the phrase based MT system we add two features (semantic similarity scores) to the bilingual phrase pairs. Since we need the vector representation of a phrase, we employ the works of (Mitchell and Lapata, 2008) on compositional semantics (adding the vectors) to compute the phrase representation. For a give phrase pair (*s,t*), we transform each constituent word of the source phrase *'s'* to the target word space and add the the transformed word embedding to the resultant source vector. We ignore the word if it does not occur in the word embeddings vocabulary. Similarly, we compute the phrase representation of the target phrase *'t'* by simply adding the word vectors to the resultant target vector. We then compute the cosine similarity between the two vectors which acts as a feature for the MT decoder. We also include the similarity score of transforming the target word phrase to source phrase as another feature. The phrase table is tuned with the previously used development data (development set used for tuning baseline MT system) using the MERT algorithm to compute the weight parameters for the baselines features and semantic similarity features.

5 Results and Discussion

The results of word similarity scores on the test set (bilingual dictionary words section 4.3) are presented in table 3 using the computed transformation matrix for English to Hindi. The similarity scores are continuously decreasing with increase in dimension, which shows that the proposed approach works better at lower dimensions for word similarity task. The word2Vec model is performing better than the GloVe model on word-similarity task. Within the same model the word2vec model with context window of five performs better than the model with context window of seven, while it is opposite for the GloVe model.

Dimension	Eng-Hin	Hin-Eng
50	19.69	**22.97**
100	19.39	22.69
150	19.58	**22.90**
200	19.80	**23.31**
250	**20.05**	**23.15**
300	**20.18**	**23.21**
400	19.75	**23.34**
500	19.37	**23.36**

Table 4: BLEU score of system using Word2Vec model with a context window of 5.

The results of our experiments (on the same test data used for evaluating the baseline MT systems) with varying dimensionality and context window are presented in table 4, 5, 6 and 7. Each of the bold marked values in the tables indicate an increase in BLEU score over the baseline. The figure 1, 2, 3 and 4 presents the comparison of BLEU score for each of the model. The highest BLEU score achieved for English-Hindi translation system is **20.53** (increase of **0.64** BLEU score over the baseline) using GloVe model with a 500 dimension vector and a context window of 5, whereas the highest score for Hindi-English system is **23.56** (increase of **0.74** BLEU score over the baseline) using word2Vec model and context window of 7. It is quite interesting to note that the increasing dimensionality and context window does not ensure increasing BLEU scores. It is evident that at a certain dimensionality the decoder algorithm (combining feature scores using log-linear model) can start distinguishing between the good and bad translations. The Hindi-English system shows improvements for almost all the cases, whereas English-Hindi system does not show similar behavior. Though the word similarity scores indicates better performance at lower dimensions, the MT experiments BLEU scores does follow the same trend. Since this language pair has not been widely explored, the results on word similarity and MT scores are not directly comparable to the earlier proposed methods.

Dimension	Eng-Hin	Hin-Eng
50	19.81	**22.93**
100	19.85	**23.01**
150	19.55	**23.29**
200	**20.37**	**22.85**
250	**20.36**	**23.16**
300	**20.02**	22.32
400	19.47	**23.13**
500	19.67	**23.56**

Table 5: BLEU score of system using Word2Vec model with a context window of 7.

Dimension	Eng-Hin	Hin-Eng
50	19.75	**23.41**
100	19.60	22.84
150	**20.28**	**23.08**
200	19.77	**22.93**
250	**20.04**	**23.30**
300	**19.97**	**23.17**
400	19.85	**22.93**
500	**20.53**	22.72

Table 6: BLEU score of system using GloVe model with a context window of 5.

Dimension	Eng-Hin	Hin-Eng
50	**20.35**	22.78
100	19.81	**23.27**
150	**20.12**	22.81
200	19.12	**23.16**
250	19.85	22.60
300	**19.88**	**23.29**
400	**20.07**	**22.83**
500	**20.01**	**23.07**

Table 7: BLEU score of system using GloVe model with a context window of 7.

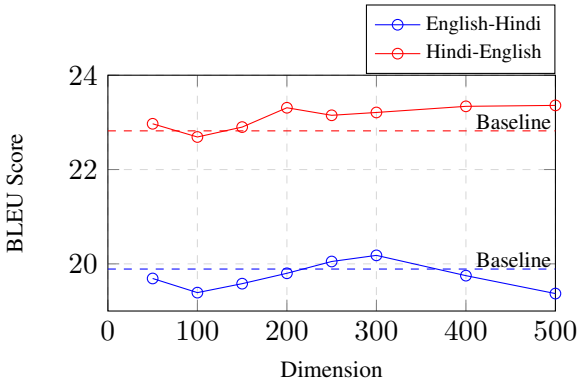

Figure 1: Plot of BLEU score variation using Word2Vec with a context window of 5

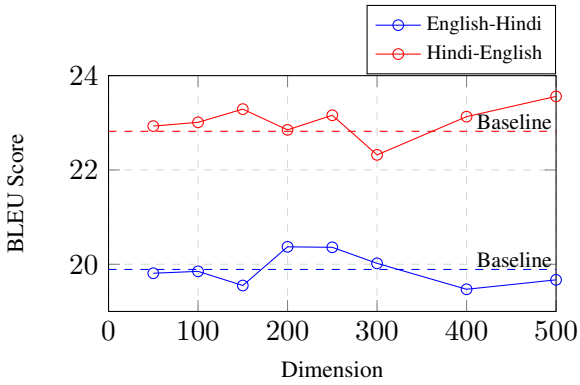

Figure 2: Plot of BLEU score variation using Word2Vec with a context window of 7

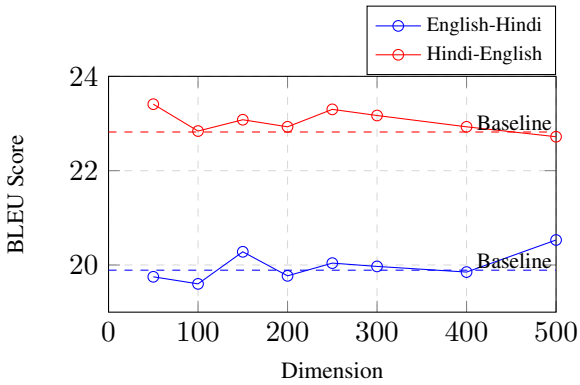

Figure 3: Plot of BLEU score variation using GloVe with a context window of 5

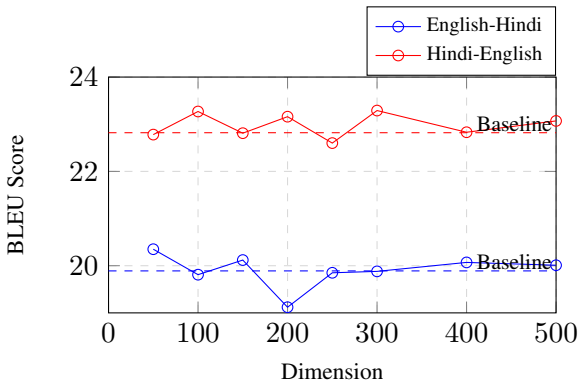

Figure 4: Plot of BLEU score variation using GloVe with a context window of 7

6 Conclusion and Future Work

In this paper we explore the use of semantic similarity between phrase pairs as features while decoding the n-best list. The bilingual word embeddings are learnt through PLS regression using a bilingual dictionary (which is an easily available resource considering low resourced language pairs as well) with limited vocabulary size. This method shows an increase in BLEU score for both English-Hindi and Hindi-English MT systems. This approach is quite effective in terms of overall complexity as the models developed by Zou (2013) and Zhang (2014) require much larger time for training.

As a part of future work, we propose the use of auto-encoders(Socher et al., 2011) to learn phrase representations as currently we are treating 'black'+'forest' and 'forest'+'black' to be having the same vector representation while semantically they are different . Since the words in one language can not be just linearly transformable to another language we will try to explore the use of feed-forward neural networks to learn non-linear transformations while minimizing the euclidean distance between the word embedding pairs. We also plan to extend the work by including the linguistic information in the word embeddings and taking the advantage of Hindi being a morphologically rich language.

References

Hervé Abdi. 2003. Partial least squares regression (pls-regression).

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2013. Learning semantic representations for the phrase translation model. *arXiv preprint arXiv:1312.0482*.

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 the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Eric Huang. 2011. Paraphrase detection using recursive autoencoder.

Girish Nath Jha. 2010. The tdil program and the indian language corpora initiative (ilci). In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA)*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL*, pages 236–244.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Haiyang Wu, Daxiang Dong, Wei He, Xiaoguang Hu, Dianhai Yu, Hua Wu, Haifeng Wang, and Ting Liu. 2014. Improve statistical machine translation with context-sensitive bilingual semantic embedding model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 142–146.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*.

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