English-to-Hindi system description for WMT 2014: Deep Source-Context Features for Moses

Marta R. Costa-jussà[1], Parth Gupta[2], Rafael E. Banchs[3] and Paolo Rosso[2]
[1]Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
[2]NLE Lab, PRHLT Research Center, Universitat Politècnica de València
[3]Human Language Technology, Institute for Infocomm Research, Singapore
[1]marta@nlp.cic.ipn.mx, [2]{pgupta,prosso}@dsic.upv.es,
[3]rembanchs@i2r.a-star.edu.sg

Abstract

This paper describes the IPN-UPV participation on the English-to-Hindi translation task from WMT 2014 International Evaluation Campaign. The system presented is based on Moses and enhanced with deep learning by means of a source-context feature function. This feature depends on the input sentence to translate, which makes it more challenging to adapt it into the Moses framework. This work reports the experimental details of the system putting special emphasis on: how the feature function is integrated in Moses and how the deep learning representations are trained and used.

1 Introduction

This paper describes the joint participation of the Instituto Politécnico Nacional (IPN) and the Universitat Politècnica de Valencia (UPV) in cooperation with Institute for Infocomm Research (I2R) on the 9th Workshop on Statistical Machine Translation (WMT 2014). In particular, our participation was in the English-to-Hindi translation task.

Our baseline system is an standard phrase-based SMT system built with Moses (Koehn et al., 2007). Starting from this system we propose to introduce a source-context feature function inspired by previous works (R. Costa-jussà and Banchs, 2011; Banchs and Costa-jussà, 2011). The main novelty of this work is that the source-context feature is computed in a new deep representation.

The rest of the paper is organized as follows. Section 2 presents the motivation of this semantic feature and the description of how the source context feature function is added to Moses. Section 3 explains how both the latent semantic indexing and deep representation of sentences are used to better compute similarities among source contexts. Section 4 details the WMT experimental framework and results, which proves the relevance of the technique proposed. Finally, section 5 reports the main conclusions of this system description paper.

2 Integration of a deep source-context feature function in Moses

This section reports the motivation and description of the source-context feature function, together with the explanation of how it is integrated in Moses.

2.1 Motivation and description

Source context information in the phrase-based system is limited to the length of the translation units (phrases). Also, all training sentences contribute equally to the final translation.

We propose a source-context feature function which measures the similarity between the input sentence and all training sentences. In this way, the translation unit should be extended from $source|||target$ to $source|||target|||trainingsentence$, with the $trainingsentence$ the sentence from which the $source$ and $target$ phrases were extracted. The measured similarity is used to favour those translation units that have been extracted from training sentences that are similar to the current sentence to be translated and to penalize those translation units that have been extracted from unrelated or dissimilar training sentences as shown in Figure 2.1.

In the proposed feature, sentence similarity is measured by means of the cosine distance in a reduced dimension vector-space model, which is constructed either by means of standard latent semantic analysis or using deep representation as decribed in section 3.

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 79–83,
Baltimore, Maryland USA, June 26–27, 2014.

S1: we could not book the room in time
T1: हम समय में टिकट आरक्षित नहीं कर सकें

S2: I want to write the book in time
T2: मैं समय में किताब लिखना चाहता हूँ

Input: i am reading a nice book

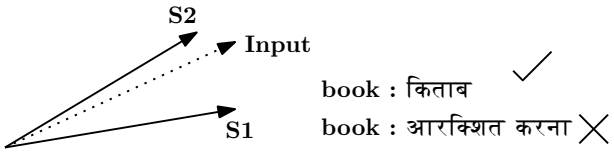

Figure 1: Illustration of the method

2.2 Integration in Moses

As defined in the section above and, previously, in (R. Costa-jussà and Banchs, 2011; Banchs and Costa-jussà, 2011), the value of the proposed source context similarity feature depends on each individual input sentence to be translated by the system. We are computing the similarity between the source input sentence and all the source training sentences.

This definition implies the feature function depends on the input sentence to be translated. To implement this requirement, we followed our previous implementation of an off-line version of the proposed methodology, which, although very inefficient in the practice, allows us to evaluate the impact of the source-context feature on a state-of-the-art phrase-based translation system. This practical implementation follows the next procedure:

1. Two sentence similarity matrices are computed: one between sentences in the development and training sets, and the other between sentences in the test and training datasets.

2. Each matrix entry m_{ij} should contain the similarity score between the i^{th} sentence in the training set and the j^{th} sentence in the development (or test) set.

3. For each sentence s in the test and development sets, a phrase pair list L_S of all potential phrases that can be used during decoding is extracted from the aligned training set.

4. The corresponding source-context similarity values are assigned to each phrase in lists L_S according to values in the corresponding similarity matrices.

5. Each phrase list L_S is collapsed into a phrase table T_S by removing repetitions (when removing repeated entries in the list, the largest value of the source-context similarity feature is retained).

6. Each phrase table is completed by adding standard feature values (which are computed in the standard manner).

7. Moses is used on a sentence-per-sentence basis, using a different translation table for each development (or test) sentence.

3 Representation of Sentences

We represent the sentences of the source language in the latent space by means of linear and non-linear dimensionality reduction techniques. Such models can be seen as topic models where the low-dimensional embedding of the sentences represent conditional latent topics.

3.1 Deep Autoencoders

The non-linear dimensionality reduction technique we employ is based on the concept of deep learning, specifically deep autoencoders. Autoencoders are three-layer networks (input layer, hidden layer and output layer) which try to learn an identity function. In the neural network representation of autoencoder (Rumelhart et al., 1986), the visible layer corresponds to the input layer and hidden layer corresponds to the latent features. The autoencoder tries to learn an abstract representation of the data in the hidden layer in such a way that minimizes reconstruction error. When the dimension of the hidden layer is sufficiently small, autoencoder is able to generalise and derive powerful low-dimensional representation of data. We consider bag-of-words representation of text sentences where the visible layer is binary feature vector ($\mathbf{v}$) where v_i corresponds to the presence or absence of i^{th} word. We use binary restricted Boltzmann machines to construct an autoencoder as shown in (Hinton et al., 2006). Latent representation of the input sentence can be obtained as shown below:

$$p(\mathbf{h}|\mathbf{v}) = \sigma(W * \mathbf{v} + \mathbf{b}) \quad (1)$$

where W is the symmetric weight matrix between visible and hidden layer and $\mathbf{b}$ is hidden layer bias vector and $\sigma(x)$ is sigmoid logistic function $1/(1 + \exp(-x))$.

Autoencoders with single hidden layer do not have any advantage over linear methods like PCA (Bourlard and Kamp, 1988), hence we consider deep autoencoder by stacking multiple RBMs on top of each other (Hinton and Salakhutdinov, 2006). The autoencoders have always been difficult to train through back-propagation until greedy layerwise pre-training was found (Hinton and Salakhutdinov, 2006; Hinton et al., 2006; Bengio et al., 2006). The pre-training initialises the network parameters in such a way that fine-tuning them through back-propagation becomes very effective and efficient (Erhan et al., 2010).

3.2 Latent Semantic Indexing

Linear dimensionality reduction technique, latent semantic indexing (LSI) is used to represent sentences in abstract space (Deerwester et al., 1990). The term-sentence matrix (**X**) is created where x_{ij} denotes the occurrence of i^{th} term in j^{th} sentence. Matrix **X** is factorized using singular value decomposition (SVD) method to obtain top m principle components and the sentences are represented in this m dimensional latent space.

4 Experiments

This section describes the experiments carried out in the context of WMT 2014. For English-Hindi the parallel training data was collected by Charles University and consisted of 3.6M English words and 3.97M Hindi words. There was a monolingual corpus for Hindi comming from different sources which consisted of 790.8M Hindi words. In addition, there was a development corpus of news data translated specifically for the task which consisted of 10.3m English words and 10.1m Hindi words. For internal experimentation we built a test set extracted from the training set. We selected randomly 429 sentences from the training corpus which appeared only once, removed them from training and used them as internal test set. Monolingual Hindi corpus was used to build a larger language model. The language model was computed doing an interpolation of the language model trained on the Hindi part of the bilingual corpus (3.97M words) and the language model trained on the monolingual Hindi corpus (790.8M words). Interpolation was optimised in the development set provided by the organizers. Both language models interpolated were 5-grams using Kneser-Ney smoothing.

The preprocessing of the corpus was done with the standard tools from Moses. English was lowercased and tokenized. Hindi was tokenized with the simple tokenizer provided by the organizers. We cleaned the corpus using standard parameters (i.e. we keep sentences between 1 and 80 words of length).

For training, we used the default Moses options, which include: the *grow-diag-final and* word alignment symmetrization, the lexicalized reordering, relative frequencies (conditional and posterior probabilities) with phrase discounting, lexical weights and phrase bonus for the translation model (with phrases up to length 10), a language model (see details below) and a word bonus model. Optimisation was done using the MERT algorithm available in Moses. Optimisation is slow because of the way integration of the feature function is done that it requires one phrase table for each input sentence.

During translation, we dropped unknown words and used the option of minimum bayes risk decoding. Postprocessing consisted in de-tokenizing Hindi using the standard detokenizer of Moses (the English version).

4.1 Autoencoder training

The architecture of autoencoder we consider was n-500-128-500-n where n is the vocabulary size. The training sentences were stemmed, stopwords were removed and also the terms with sentences frequency[1] less than 20 were also removed. This resulted in vocabulary size n=7299.

The RBMs were pretrained using Contrastive Divergence (CD) with step size 1 (Hinton, 2002). After pretraining, the RBMs were stacked on top of each other and unrolled to create deep autoencoder (Hinton and Salakhutdinov, 2006). During the fine-tuning stage, we backpropagated the reconstruction error to update network parameters. The size of mini-batches during pretraining and fine-tuning were 25 and 100 respectively. Weight decay was used to prevent overfitting. Additionally, in order to encourage sparsity in the hidden units, Kullback-Leibler sparsity regularization was used. We used GPU[2] based implementation of autoencoder to train the models which took around 4.5 hours for full training.

[1] total number of training sentences in which the term appears

[2] NVIDIA GeForce GTX Titan with Memory 5 GiB and 2688 CUDA cores

4.2 Results

Table 1 shows the improvements in terms of BLEU of adding deep context over the baseline system for English-to-Hindi (En2Hi) over development and test sets. Adding source-context information using deep learning outperforms the latent semantic analysis methodology.

	En2Hi	
	Dev	Test
baseline	9.42	14.99
+lsi	**9.83**	**15.12**
+deep context	**10.40**†	**15.43**†

Table 1: BLEU scores for En2Hi translation task.. † depicts statistical significance (p-value$<$0.05).

Our source-context feature function may be more discriminative in a task like English-to-Hindi where the target language has a larger vocabulary than the source one.

Table 2 shows an example of how the translation is improving in terms of lexical semantics which is the goal of the methodology presented in the paper. The most frequent sense of word *cry* is रोना, which literally means "to cry" while the example in Table 2 refers to the sense of *cry* as चीख, which means to *scream*. Our method could identify the context and hence the source context feature (scf) of the unit cry|||चीख is higher than for the unit scf(cry|||रोना) as shown in Table 3 and for this particular input sentence.

5 Conclusion

This paper reports the IPN-UPV participation in the WMT 2014 Evaluation Campaign. The system is Moses-based with an additional feature function based on deep learning. This feature function introduces source-context information in the standard Moses system by adding the information of how similar is the input sentence to the different training sentences. Significant improvements over the baseline system are reported in the task from English to Hindi.

As further work, we will implement our feature function in Moses using suffix arrays in order to make it more efficient.

System	**Translation**
Source	soft cry from the depth
Baseline	गहराइयों से मूलायम रोने लगते
+deep context	गहराइयों से मूलायम **चीख**
Reference	गहराइयों से कोमल **चीख**

Table 2: Manual analysis of a translation output.

	cp	***pp***	***scf***
cry|||रोना	0.23	0.06	0.85
cry|||चीख	0.15	0.04	0.90

Table 3: Probability values (conditional, cp, and posterior, pp, as standard features in a phrase-based system) for the word *cry* and two Hindi translations.

Acknowledgements

This work has been supported in part by Spanish Ministerio de Economía y Competitividad, contract TEC2012-38939-C03-02 as well as from the European Regional Development Fund (ERDF/FEDER). The work of the second and fourth authors is also supported by WIQ-EI (IRSES grant n. 269180) and DIANA-APPLICATIONS (TIN2012-38603-C02-01) project and VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

References

Rafael E. Banchs and Marta R. Costa-jussà. 2011. A semantic feature for statistical machine translation. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, SSST-5, pages 126–134.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160.

Hervé Bourlard and Yves Kamp. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, September.

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.

Dumitru Erhan, Yoshua Bengio, Aaron C. Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660.

Geoffrey Hinton and Ruslan Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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*, ACL '07, pages 177–180.

Marta R. Costa-jussà and Rafael E. Banchs. 2011. The bm-i2r haitian-créole-to-english translation system description for the wmt 2011 evaluation campaign. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 452–456, Edinburgh, Scotland, July. Association for Computational Linguistics.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.