A Discriminative Training Procedure for Continuous Translation Models

† *Quoc-Khanh Do, † *Alexandre Allauzen and *François Yvon
† Université Paris-Sud, Orsay, France
* LIMSI/CNRS, Orsay, France
firstname.surname@limsi.fr

Abstract

Continuous-space translation models have recently emerged as extremely powerful ways to boost the performance of existing translation systems. A simple, yet effective way to integrate such models in inference is to use them in an N-best rescoring step. In this paper, we focus on this scenario and show that the performance gains in rescoring can be greatly increased when the neural network is trained jointly with all the other model parameters, using an appropriate objective function. Our approach is validated on two domains, where it outperforms strong baselines.

1 Introduction

Over the past few years, research on neural networks (NN) architectures for Natural Language Processing has been rejuvenated. Boosted by early successes in language modelling for speech recognition (Schwenk, 2007; Le et al., 2011), NNs have since been successufully applied to many other tasks (Socher et al., 2013; Huang et al., 2012; Yang et al., 2013). In particular, these techniques have been applied to Statistical Machine Translation (SMT), first to estimate continuous-space translation models (CTMs) (Schwenk et al., 2007; Le et al., 2012; Devlin et al., 2014), and more recently to implement end-to-end translation systems (Cho et al., 2014; Sutskever et al., 2014).

In most SMT settings, CTMs are used as an additional feature function in the log-linear model, and are conventionally trained by maximizing the regularized log-likelihood on some parallel training corpora. Since this objective function requires to normalize scores, several alternative training objectives have recently been proposed to speed up training and inference, a popular and effective choice being the Noise Contrastive Estimation (NCE) introduced in (Gutmann and Hyvärinen, 2010). In any case, NN training is typically performed (a) in isolation from the other components of the SMT system and (b) using a criterion that is unrelated to the actual performance of the SMT system (as measured for instance by BLEU). It is therefore likely that the resulting NN parameters are sub-optimal with respect to their intended use.

In this paper, we study an alternative training regime aimed at addressing problems (a) and (b). To this end, we propose a new objective function used to discriminatively train or adapt CTMs, along with a training procedure that enables to take the other components of the system into account. Our starting point is a non-normalized extension of the n-gram CTM of (Le et al., 2012) that we briefly restate in section 2. We then introduce our objective function and the associated optimization procedure in section 3. As will be discussed, our new training criterion is inspired both from max-margin methods (Watanabe et al., 2007) and from pair-wise ranking (PRO) (Hopkins and May, 2011; Simianer et al., 2012). This proposal is evaluated in an N-best rescoring step, using the framework of n-gram-based systems, within which they integrate seamlessly. Note, however that it could be used with any phrase-based system. Experimental results for two translation tasks (section 4) clearly demonstrate the benefits of using discriminative training on top of an NCE-trained model, as it almost doubles the performance improvements of the rescoring step in all settings.

2 n-gram-based CTMs

The n-gram-based approach in Machine Translation is a variant of the phrase-based approach (Zens et al., 2002). Introduced in (Casacuberta and Vidal, 2004), and extended in (Mariño et al., 2006; Crego and Mariño, 2006), this approach is based on a specific factorization of the joint probability of parallel sentence pairs, where the

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1046–1052,
Lisbon, Portugal, 17-21 September 2015.

source sentence has been reordered beforehand.

2.1 n-gram-based Machine Translation

Let $(\mathbf{s}, \mathbf{t})$ denote a sentence pair made of a source $\mathbf{s}$ and target $\mathbf{t}$ sides. This sentence pair is decomposed into a sequence of L bilingual units called *tuples* defining a joint segmentation. In this framework, tuples constitute the basic translation units: like phrase pairs, they represent a matching between a source and a target chunk . The joint probability of a *synchronized* and *segmented* sentence pair can be estimated using the n-gram assumption. During training, the segmentation is obtained as a by-product of source reordering, (see (Crego and Mariño, 2006) for details). During the inference step, the SMT decoder will compute and output the best derivation in a small set of pre-defined reorderings.

Note that the n-gram translation model manipulates bilingual tuples. The underlying set of events is thus much larger than for word-based models, while the training data (parallel corpora) are typically order of magnitude smaller than monolingual resources. As a consequence, data sparsity issues for such models are particularly severe. Effective workarounds consist in factorizing the conditional probabitily of tuples into terms involving smaller units: the resulting model thus splits bilingual phrases in two sequences of respectively source and target words, synchronised by the tuple segmentation. Such bilingual word-based n-gram models were initially described in (Le et al., 2012). We assume here a similar decomposition.

2.2 Neural Architectures

The estimation of n-gram probabilities can be performed via multi-layer NN structures, as described in (Bengio et al., 2003; Schwenk, 2007) for a monolingual language model. The standard *feed-forward* structure is used to estimate the translation models sketched in the previous section. We give here a brief description, more details are in (Le et al., 2012): first, each context word is projected into language dependent continuous spaces, using two projection matrices for the source and target languages. The continuous representations are then concatenated to form the representation of the context, which is used as input for a feed-forward NN predicting a target word.

In such architecture, the size of output vocabulary is a bottleneck when normalized distributions are expected. Various workarounds have been proposed, relying for instance on a structured output layer using word-classes (Mnih and Hinton, 2008; Le et al., 2011). A more effective alternative, which however only delivers *quasi-normalized* scores, is to train the network using the *Noise Contrastive Estimation* or NCE (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012). This technique is readily applicable for CTMs and has been adopted here. We therefore assume that the NN outputs a positive score $\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c})$ for each word w given its context $\mathbf{c}$; this score is simply computed as $\mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c}) = \exp(\mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c}))$, where $\mathbf{a}_{\boldsymbol{\theta}}(w, \mathbf{c})$ is the activation at the output layer; $\boldsymbol{\theta}$ denotes all the network free parameters.

3 Discriminative Training of CTMs

In SMT, the primary role of CTMs is to help the system in ranking a set of hypotheses so that the top scoring hypotheses correspond to the best translations, where quality is measured using automatic metrics such as BLEU (Papineni et al., 2002). Given the computational burden of continuous models, the prefered use of CTMs is to rescore a list of N-best hypotheses, a scenario we favor here; note that their integration in a first pass search is also possible (Niehues and Waibel, 2012; Vaswani et al., 2013; Devlin et al., 2014). The important point is to realize that the CTM score will in any case be composed with several scores computed by other components: reordering model(s), monolingual language model(s), etc. In this section, we propose a discriminative training framework which implements a tight integration of the CTM with the rest of the system.

3.1 A Discriminative Training Framework

The decoder generates a list of N hypotheses for each source sentence $\mathbf{s}$. Each hypothesis $\mathbf{h}$ is composed of a target sentence $\mathbf{t}$ along with its associated derivation and is evaluated as follows:

$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = \sum_{k=1}^{M} \lambda_k f_k(\mathbf{s}, \mathbf{h}) + \lambda_{M+1} f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}),$$

where M conventional feature functions[1] $f_1 ... f_M$, estimated during the training phase, are scaled by coefficients $\lambda_1 ... \lambda_M$. The introduction of a continuous model during the rescoring step is implemented by adding the feature $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$, which ac-

[1]The functions used in our experiments are similar to the ones used in other phrase-based systems (Crego et al., 2011).

Algorithm 1 Joint optimization of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$

```
Init. of θ and λ
for each iteration do
    for P mini-batch do                    ▷ λ is fixed
        Compute the sub-gradient of L(θ) for each sentence s in the mini-batch
        Update θ
    end for
    Update λ on development set            ▷ θ is fixed
end for
```

cumulates, over all contexts $\mathbf{c}$ and word w, the CTM log-score $\log \mathbf{b}_{\boldsymbol{\theta}}(w, \mathbf{c})$.

$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}$ depends both on the NN parameters $\boldsymbol{\theta}$ and on the log-linear coefficients $\boldsymbol{\lambda}$. We propose to train these two sets of parameters, by alternatively updating $\boldsymbol{\theta}$ through SGD on the training corpus, and updating $\boldsymbol{\lambda}$ using conventional algorithms on the development data. This procedure, which has also been adopted in recent studies (e.g. (He and Deng, 2012; Gao and He, 2013)) is sketched in algorithm 1. In practice, the training data is successively divided into mini-batches of 128 sentences. Each mini-batch is used to compute the sub-gradient of the training criterion (see section 3.2) and to update $\boldsymbol{\theta}$. After each training iteration of the CTM, $\boldsymbol{\lambda}$s are retuned on the development set; we use here the *K-Best Mira* algorithm of Cherry and Foster (2012) as implemented in MOSES.[2]

3.2 Loss function

The training criterion considered here draws inspiration both from max-margin methods (Watanabe et al., 2007) and from the pair-wise ranking (PRO) (Hopkins and May, 2011; Simianer et al., 2012). The choice of a ranking loss seems to be the most appropriate in our setting; as in many recent studies on discriminative training for MT (e.g. (Chiang, 2012; Flanigan et al., 2013)), the integration of the translation metric into the loss function is critical to obtain parameters that will yield good translation performance.

Translation hypotheses $\mathbf{h}_i$ are scored using a sentence-level approximation of BLEU denoted $SBLEU(\mathbf{h}_i)$. Let r_i be the rank of hypothesis $\mathbf{h}_i$ when hypotheses are sorted according to their sentence-level BLEU. *Critical hypotheses* are defined as follows:[3]

$$\mathcal{C}^{\alpha}_{\delta}(\mathbf{s}) = \{(i,k) : 1 \leq k, i \leq N, r_k - r_i \geq \delta, \\ \Delta_{i,k} G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) < \alpha \Delta_{i,k} SBLEU(\mathbf{h}).$$

A pair of hypotheses is thus deemed critical when a large difference in $SBLEU$ is not reflected by the difference of scores, which falls below a threshold. This threshold is defined by the difference between their sentence-level BLEU, multiplied by α. Our loss function $\mathcal{L}(\boldsymbol{\theta})$ is defined with respect to this critical set and can be written as:[4]

$$\sum_{(i,k) \in \mathcal{C}^{\alpha}_{\delta}(\mathbf{s})} \alpha \Delta_{i,k} SBLEU(\mathbf{h}) - \Delta_{i,k} G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i)$$

Initialization is an important issue when optimizing NN. Moreover, our training procedure depends heavily on the log-linear coefficients $\boldsymbol{\lambda}$. To initialize $\boldsymbol{\theta}$, preliminary experiments (Do et al., 2014; Do et al., 2015) show that it is more efficient to start from a NN pre-trained using NCE, while the discriminative loss is used only in a fine-tuning phase. Given the pre-trained CTM's scores, we initialize $\boldsymbol{\lambda}$ by optimizing it on the development set. This strategy forces the training of $\boldsymbol{\theta}$ to focus on errors made by the system as a whole.

4 Experiments

4.1 Tasks and Corpora

The discriminative optimization framework is evaluated both in a training and in an adaptation scenario. In the *training* scenario, the CTM is trained on the same parallel data as the one used for the baseline system. In the *adaptation* scenario, large out-of-domain corpora are used to train the baseline SMT system, while the CTM is trained on a much smaller, in-domain corpus and only serves for rescoring. An intermediate situation (*partial training*) is when only a fraction of the training data is re-used to estimate the CTM: this situation is interesting because it allows us to train the CTM much faster than in the training scenario.[5]

Two domains are investigated. For the TED Talkstask[6] the only parallel in-domain data contains 180K sentence pairs; the out-of-domain

[2] http://www.statmt.org/moses/

[3] $\Delta_{i,k}$ denotes the difference of values (for SBLEU or $G_{\boldsymbol{\lambda},\boldsymbol{\theta}}$) between hypthoses h_i and h_k.

[4] This is for one single training sample.

[5] The discriminative training step also uses the development data.

[6] http://workshop2014.iwslt.org/

	dev	test	train
Training scenario			
Baseline Ncode on TED	28.1	32.3	**65.6**
Baseline + CTM NCE	28.9	33.1	64.1
Baseline + CTM discriminative	**29.0**	**33.5**	64.9
Adaptation scenario			
Baseline Ncode on WMT	28.5	32.0	33.3
Baseline + CTM NCE	29.2	33.0	34.9
Baseline + CTM discriminative	**29.8**	**33.9**	**35.8**

Table 1: BLEU scores for the TED Talkstasks.

	dev	test	train
Partial training scenario			
Baseline Ncode	40.4	37.4	45.8
Baseline + CTM NCE	40.8	38.1	45.2
Baseline + CTM discriminative	**41.8**	**38.8**	**46.0**
"Adaptation" scenario			
Baseline Ncode	39.8	37.2	39.4
Baseline + CTM NCE	41.2	38.2	40.4
Baseline + CTM discriminative	**41.8**	**38.9**	**41.5**

Table 2: BLEU scores for the medical tasks.

data is much larger and contains all corpora allowed in the translation shared task of WMT'14 (English-French), amounting to 12M parallel sentences. The second task is the medical translation task of WMT'14[7] (English to French) for which we use all authorized corpora. The Patent-Abstract corpus, made of 200K parallel sentence pairs, is used either for adaptation or partial training for the CTM. Experimental results are reported on official evaluation sets, as well as on the CTM training set.

All translation systems are based on the open source implementation[8] of the bilingual n-gram approach to MT. For the NN structure, each vocabulary's word is projected into a 500-dimension space followed by two hidden layers of 1000 and 500 units. For the discriminative training and adaptation tasks, baseline SMT systems are used to generate respectively 600 and 300 best hypotheses for each sentence of the in-domain corpus. [9]

4.2 Experimental results

Results in Table 1 measure the impact of discriminative training on top of an NCE-trained model for the two TED Talks conditions. In the adaptation task, the discriminative training of the CTM gives a large improvement of 0.9 BLEU score over the CTM only trained with NCE and 1.9 over the baseline system. However, for the training scenario, these gains are reduced respectively to 0.4 and 1.2 BLEU points. The BLEU scores (in the **train** column) measured on the N-best lists used to train the CTM provide an explanation for this difference: in training, the N-best lists contain hypotheses with an overoptimistic BLEU score, to be compared with the ones observed on unseen data. As a result, adding the CTM significantly worsens the performance on the discriminative training data, contrarily to what is observed on the development and test sets. Even if the results of these two conditions cannot be directly compared (the baselines are different), it seems that the proposed discriminative training has a greater impact on performance in the adaptation scenario, even though the out-of-domain system initially yields lower BLEU scores.

The medical translation task represents a different situation, in which a large-scale system is built from multiples but domain-related corpora, among which, one is used to train the CTM. Nevertheless, results reported in Table 2 exhibit a similar trend. For both conditions, the discriminative training gives a significant improvement, up to 0.7 BLEU score over the one only trained with NCE and up to 1.7 over the baseline system. Arguably, the difference between the two conditions is much smaller than what was observed with the TED Talks task, due to the fact that the Patent-Abstract corpus used to discriminatively train the CTM only corresponds to a small subset of the parallel data. However, the best strategy seems, here again, to exclude the data used for the CTM from the data used to train the baseline system.

5 Related work

It is important to notice that similar discriminative methods have been used to train phrase table's scores (He and Deng, 2012; Gao and He, 2013; Gao et al., 2014), or a recurrent NNLM (Auli and Gao, 2014). In recent studies, the authors tend to limit the number of iterations to 1 (Gao et al., 2014; Auli and Gao, 2014), while we still advocate the general iterative procedure sketched in Algo. 1. Initialization is also an important issue when optimizing NN. In this work, we initialize CTM's parameters by using a pre-training procedure based on the model's probabilistic in-

[7] www.statmt.org/wmt14/medical-task/

[8] ncode.limsi.fr/

[9] The threshold δ is set to 250 for 300-best and to 500 for 600-best lists, while α is set empirically.

terpretation and NCE algorithm to produce *quasi-normalized scores*, while similar work in (Auli and Gao, 2014) only uses *un-normalized scores*. The initial values of $\boldsymbol{\lambda}$ also needs some investigation. Gao et al. (2014) and Auli and Gao (2014) initialize λ_{M+1} to 1, and normalize all other coefficients; here we initialize $\boldsymbol{\lambda}$ by optimizing it on the development set using the pre-trained CTM's scores. This strategy forces the training of $\boldsymbol{\theta}$ to focus on errors made by the system as a whole. The fundamental difference of this work hence lays in the use of the ranking loss described in Section 3.2, whereas previous works use expected BLEU loss. We plan a systematic comparison between these two criteria, along with some other discriminative losses in a future work.

About the CTM's structure, our used model is based on the feed-forward CTM described in (Le et al., 2012) and extended in (Devlin et al., 2014). This structure, though simple, have been shown to achieve impressive results, and with which efficient tricks are available to speed up both training and inference. While models in (Le et al., 2012) employ a structured output layer to reduce *softmax* operation's cost, we prefer the NCE *self-normalized* output which is very efficient both in training and inference. Another form of self-normalization is presented in (Devlin et al., 2014) but does not seem to have fast training. Finally, although N-best rescoring is used in this work to facilitate the discriminative training, other CTM's integration into SMT systems exist, such as lattice reranking (Auli et al., 2013) or direct decoding with CTM (Niehues and Waibel, 2012; Devlin et al., 2014; Auli and Gao, 2014).

6 Conclusions

In this paper, we have proposed a new discriminative training procedure for continuous-space translation models, which correlates better with translation quality than conventional training methods. This procedure has been validated using an n-gram-based CTM, but the general idea could be applied to other continuous models which compute a score for each translation hypothesis. The core of the method lays in the definition of a new objective function inspired both from max-margin and Pairwise Ranking approach in MT, which enables us to effectively integrate the CTM into the SMT system through N-best rescoring. A major difference with most past efforts along these lines is the joint training of the CTM and the log-linear parameters. In all our experiments, discriminative training, when applied on a CTM initially trained with NCE, yields substantial performance gains.

Acknowledgments

This work has been partly funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

References

Michael Auli and Jianfeng Gao. 2014. Decoder integration and expected bleu training for recurrent neural network language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 136–142.

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1044–1054.

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

Francesco Casacuberta and Enrique Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(3):205–225.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 427–436.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research*, 13(1):1159–1187.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar.

Josep M. Crego and José B. Mariño. 2006. Improving statistical MT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

Josep M. Crego, François Yvon, and José B. Mariño. 2011. N-code: an open-source bilingual N-gram SMT toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, MD.

Quoc-Khanh Do, Alexandre Allauzen, and François Yvon. 2014. Discriminative adaptation of continuous space translation models. In *International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, USA.

Quoc-Khanh Do, Alexandre Allauzen, and François Yvon. 2015. Apprentissage discriminant des modèles continus de traduction. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles*, pages 267–278, Caen, France, June. Association pour le Traitement Automatique des Langues.

Jeffrey Flanigan, Chris Dyer, and Jaime Carbonell. 2013. Large-scale discriminative training for statistical machine translation using held-out line search. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 248–258, Atlanta, Georgia.

Jianfeng Gao and Xiaodong He. 2013. Training mrf-based phrase translation models using gradient ascent. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 450–459, Atlanta, Georgia.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yeh Whye Teh and Mike Titterington, editors, *Proceedings of th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 297–304.

Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 292–301.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK.

Eric Huang, Richard Socher, Christopher Manning, and Andrew 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 (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 5524–5527.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 39–48, Montréal, Canada.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrick Lambert, José A.R. Fonollosa, and Marta R. Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 1081–1088.

Andriy Mnih and Yeh Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference of Machine Learning (ICML)*.

Jan Niehues and Alex Waibel. 2012. Continuous space language models using restricted Boltzmann machines. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, pages 164–170, Hong-Kong, China.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Holger Schwenk, Marta R. Costa-jussa, and Jose A. R. Fonollosa. 2007. Smooth bilingual n-gram translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 430–438, Prague, Czech Republic.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492–518.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11–21.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 455–465, Sofia, Bulgaria.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, NIPS*27, pages 3104–3112, Montréal, Canada.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1387–1392, Seattle, Washington, USA.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 166–175, Sofia, Bulgaria.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *KI '02: Proceedings of the 25th Annual German Conference on AI*, pages 18–32, London, UK. Springer-Verlag.