ListNet-based MT Rescoring

[†]Jan Niehues, [*]Quoc Khanh Do, [*]Alexandre Allauzen and [†]Alex Waibel
[†]Karlsruhe Institute of Technology, Karlsruhe, Germany
[*]LIMSI-CNRS, Orsay, France
[†]firstname.surname@kit.edu [*]surname@limsi.fr

Abstract

The log-linear combination of different features is an important component of SMT systems. It allows for the easy integartion of models into the system and is used during decoding as well as for n-best list rescoring. With the recent success of more complex models like neural network-based translation models, n-best list rescoring attracts again more attention. In this work, we present a new technique to train the log-linear model based on the ListNet algorithm. This technique scales to many features, considers the whole list and not single entries during learning and can also be applied to more complex models than a log-linear combination.

Using the new learning approach, we improve the translation quality of a large-scale system by 0.8 BLEU points during rescoring and generate translations which are up to 0.3 BLEU points better than other learning techniques such as MERT or MIRA.

1 Introduction

Nowadays, statistical machine translation is the most promising approach to translate from one natural language into another one, when sufficient training data is available. While there are several powerful approaches to model the translation process, nearly all of them rely on a log-linear combination of different models. This approach allows the system an easy integration of additional models into the translation process and therefore a great flexibility to address the various issues and the different language pairs.

The log-linear model is used during decoding and for n-best list rescoring. Recently, the success of rich but computationally complex models, such as neural network based translation models (Le et al., 2012), leads to an increased interest in rescoring. It was shown that the n-best list rescoring is an easy and efficient way to integrate complex models.

From a machine learning perspective the log-linear model is used to solve a ranking problem. Given a list of candidates associated with different features, we need to find the best ranking according to a reference ranking. In machine translation, this ranking is, for example, given by an automatic evaluation metric. One promising approach for this type of problems is the ListNet algorithm (Cao et al., 2007), which has already been applied successfully to the information retrieval task. Using this algorithm it is possible to train many features. In contrast to other algorithms, which work only on single pairs of entries, it considers the whole list during learning. Furthermore, in addition to train the weights of a linear combination, it can be used for more complex models such as neural networks.

In this paper, we present an adaptation of this algorithm to the task of machine translation. Therefore, we investigate different methods to normalize the features and adapt the algorithm to directly optimize a machine translation metric. We used the algorithm to train a rescoring model and compared it to several existing training algorithms.

In the following section, we first review the related work. Afterwards, we introduce the ListNet algorithm in Section 3. The adaptation to the problem of rescoring machine translation n-best lists will be described in the next section. Finally, we will present the results on different language pairs and domains.

2 Related Work

The first approach to train the parameters of the log-linear combination model in statistical machine translation was the minimum error rate train-

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 248–255,
Lisboa, Portugal, 17-18 September 2015. ©2015 Association for Computational Linguistics.

ing (MERT) (Och, 2003). Although new methods have been presented, this is still the standard method in many machine translation systems. One problem of this technique is that it does not scale well with many features. More recently, Watanabe et al. (2007) and Chiang et al. (2008) presented a learning algorithm using the MIRA technique. A different technique, PRO, was presented in (Hopkins and May, 2011). Additionally, several techniques to maximize the expected BLEU score (Rosti et al., 2011; He and Deng, 2012) have been proposed. The ListNet algorithm, in contrast, minimizes the difference between the model and the reference ranking. All techniques have the advantage that they can scale well to many features and an intensive comparison of these methods is reported in (Cherry and Foster, 2012).

The problem of ranking is well studied in the machine learning community (Chen et al., 2009). These methods can be grouped into pointwise, pairwise and listwise algorithms. The PRO algorithm is motivated by a pairwise technique, while the work presented in this paper is based on the listwise algorithm ListNet presented in (Cao et al., 2007). Other methods based on more complex models have also been presented, for example (Liu et al., 2013), which uses an additive neural network instead of linear models.

3 ListNet

The ListNet algorithm (Cao et al., 2007) is a listwise approach to the problem of ranking. Every list of candidates that need to be ranked is used as an instance during learning. The algorithm has already been successfully applied to the task of information retrieval.

In order to use the listwise approach for learning, we need to define a loss function that considers a whole list. The idea in the ListNet algorithm is to define two probability distributions respectively on the hypothesized and reference ranking. Then a metric that compares both distributions can define the loss function. In this case, we will learn a scoring function that defines a probability distribution over the possible permutations of the candidate list which is similar to the reference ranking.

For a given set of m candidate lists $l = \{l^{(1)}, \dots, l^{(m)}\}$, each list $l^{(i)}$ contains a set of $n^{(i)}$ features vectors $x^{(i)} = \{x_1^{(i)} \dots, x_{n^{(i)}}^{(i)}\}$ associated to a set of reference scores $y^{(i)} = \{y_1^{(i)} \dots, y_{n^{(i)}}^{(i)}\}$, where $n^{(i)}$ is the number of elements in the list $l^{(i)}$. The aim is then to find a function f_ω that assigns a score to every feature vector $x_j^{(i)}$. This function is fully defined by its set of parameters ω. Using the vector of scores $z^{(i)} = \{f_\omega(x_1^{(i)}) \dots, f_\omega(x_{n^{(i)}}^{(i)})\}$ and the reference scores $y^{(i)}$, a listwise loss function must be defined to learn the function f_ω.

Since the number of permutations is $n!$ hence prohibitive, Cao et al. (2007) suggests to replace the probability distribution over all the permutations by the probability that an object is ranked first. This can be defined as:

$$P_s(j) = \frac{\exp(s_j)}{\sum_{k=1}^{n} \exp(s_k)}, \qquad (1)$$

where s_j is a score assigned to the j-th entry of the list, either $z_j^{(i)}$ or $y_j^{(i)}$. Then a loss function is defined by the cross entropy to compare the distribution of the reference ranking with the induced ranking:

$$L(y^{(i)}, z^{(i)}) = -\sum_{j=1}^{n} P_{y^{(i)}}(j) \log(P_{z^{(i)}}(j)) \qquad (2)$$

The gradient of the loss function with respect to the parameters ω can be computed as follows:

$$\begin{aligned} \Delta\omega &= \frac{\delta L(y^{(i)}, z^{(i)})}{\delta\omega} = \qquad (3) \\ &- \sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) \frac{\delta f_\omega(x_j^{(i)})}{\delta\omega} \\ &+ \frac{1}{\sum_{j=1}^{n^{(i)}} \exp(f_\omega(x_j^{(i)}))} \\ &\quad \sum_{j=1}^{n^{(i)}} \exp(f_\omega(x_j^{(i)})) \frac{\delta f_\omega(x_j^{(i)})}{\delta\omega} \end{aligned}$$

4 Rescoring

In this work, we used a log-linear model to rescore the hypothesis of the n-best lists. The log-linear model selects the hypothesis translations $\hat{e}_i$ of source sentence f_i according to Equation 4.

$$\hat{e}_i = \underset{j \in \{1 \dots n^{(i)}\}}{\operatorname{argmax}} \sum_{k=1}^{K} \omega_k h_k(e_i^j, f_i) \qquad (4)$$

K is the number of features, h_k are the different features and ω_k are the parameters of the model that need to be learned using the ListNet algorithm.

In this case, the sets of candidate lists l are the n-best lists generated for the development data. The scores $x_j^{(i)} = \{h_1(e_i^j, f_i) \dots h_K(e_i^j, f_i)\}$ are the features of the translation hypothesis ranked in position j for the sentence i. The features include conventional scores calculated during decoding, as well as additional models such as neural network translation models.

4.1 Score normalization

The scores $(x_j^{(i)})_k$ are, for example, language model log-probabilities. Since the language model probabilities are calculated as the product of several n-gram probabilities, these values are typically very small. Therefore, the log-probabilities are negative numbers with a high absolute value. Furthermore, the range of feature values may greatly differ. This can lead to problems in the calculation of $\exp(f_\omega(x_j^{(i)}))$. Therefore, we investigated two techniques to normalize the scores, feature normalization and final score normalization

In the feature normalization, all values of scores observed on the development data are rescaled into the range of $[-1, 1]$ using a linear transformation. Let $m_k = \min_{i,j}\{(x_j^{(i)})_k\}$ denote the minimum value of the feature k observed on the development set and similarly M_k for the maximum. The original scores are replaced by their rescaled version $(\hat{x}_j^{(i)})_k$ as follows:

$$(\hat{x}_j^{(i)})_k = \frac{2 * (x_j^{(i)})_k - (M_k + m_k)}{M_k - m_k} \qquad (5)$$

The same transformation based on the minimal and maximal feature values on the development data is applied to the test data.

When using the final score normalization, we normalize the resulting scores $f_\omega(x_j^{(i)})$. This is done separately for every n-best list. We calculate the highest absolute value M_i by:

$$M_i = \max_{j=1}^{n^{(i)}}(|f_\omega(x_j^{(i)})|) \qquad (6)$$

Then we use the rescaled scores denoted $\overline{f}_\omega$ and defined as follows:

$$\overline{f}_\omega(x_j^{(i)}) = f_\omega(x_j^{(i)}) * \frac{r}{M_i}, \qquad (7)$$

where r is the desired target range of possible scores.

Although both methods could be applied together, we did only use one of them, since both methods have similar effects.

If not stated differently, we use the feature normalization method in our experiments.

4.2 Metric

To estimate the weights, we need to define a probability distribution P_y associated to the reference ranking y following Euqation 1. In this work, we propose a distribution based on machine translation evaluation metrics.

The most widely used evaluation metric is BLEU (Papineni et al., 2002), which only produces a score at the corpus level. As proposed by Hopkins and May (2011), we will use a smoothed sentence-wise BLEU score to generate the reference ranking. In this work, we use the BLEU+1 score introduced by Liang et al. (2006). When using $s_j = \text{BLEU}(x_j^{(i)})$ in Equation 1, whe get the follwing defintion of the probability distribution P_y:

$$P_{y^{(i)}}(x_j^{(i)}) = \frac{\exp(\text{BLEU}(x_j^{(i)}))}{\sum_{j'=1}^{n^i} \exp(\text{BLEU}(x_{j'}^{(i)})} \qquad (8)$$

However, the raw use of BLEU+1 may lead to a very flat probability distribution, since the difference in BLEU among translation candidates in the n-best list is in general relatively small. Motivated by initial experiments, we use instead the BLEU+1 percentage of each sentence.

4.3 Training

Since the loss function defined in Equation 2 is differentiable and convex *w.r.t* the parameters ω, the stochastic gradient descent can be applied for optimization purpose. The model is trained by randomly selecting sentences from the development set and by applying batch updates after rescoring ten source sentences. The training process ends after 100,000 batches and the final model is selected according to its performance on the development data. The learning rate was empirically selected using the development data. We investigated fixed learning rates around 1 as well as dynamically updating the learning rate.

5 Evaluation

The proposed approach is evaluated in two widely known translation tasks. The first is the large scale translation task of WMT 2015 for the German–English language pair in both directions. The second is the task of translating English TED lectures into German using the data from the IWSLT 2015 evaluation campaign (Cettolo et al., 2014). The systems using the ListNet-based rescoring were submitted to this evaluation campaigns and when evaluated using the BLEU score they were all ranking within the top 3. Before discussing the results, we summarize the translation systems used for experiments along with the additionnal features that rely on continuous space translation models.

5.1 Systems

The baseline system is an in-house implementation of the phrase-based approach. The system used to generate n-best lists for the news tasks is trained on all the available training corpora of the WMT 2015 Shared Translation task. The system uses a pre-reordering technique and facilitates several translation and language models. A full system description can be found in (Cho et al., 2015). The German to English baseline system uses 19 features and the English to German systems uses 22 features. Both systems are tuned on news-test2013 which also serves to train the rescoring step using ListNet. The news-test2014 is dedicated for evaluation purpose. On both sets, 300-best lists are generated.

In addition to baseline features, we also analyze the influence of features calculated on the n-best list after decoding. Since we only need to calculate the scores for the entries in the n-best lists and not for all partial derivations considered during decoding, we can use more complex models.

For the English to German translation task, we used neural network translation models as introduced in (Le et al., 2012). This model decomposes the sequence of phrase pairs proposed by the translation system in two sequences of source and target words respectively, synchronized by the segmentation into phrase pairs. This decomposition defines four different scores to evaluate a hypothesis. In such architecture, the size of the output vocabulary is a bottleneck when normalized distributions are needed. For efficient computation, these models rely on a tree-structured output layer called SOUL (Le et al., 2011). An effective alternative, which however only delivers unnormalized scores, is to train the network using the Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012) denoted by NCE in the rest of the paper. In this work, we used these both solutions as well as their combination.

For the German to English translation task, we added a source side discriminative word lexicon (Herrmann, 2015). This model used a multi-class maximum entropy classifier for every source word to predict the translation given the context of the word. In addition, we used a neural network translation model using the technique of RBM (Restricted Boltzman Machine)-based language models (Niehues and Waibel, 2012).

The baseline system for the TED translation task uses the IWSLT 2015 training data. The system was adapted to the domain by using language model and translation model adaptation techniques. A detailed description of all models used in this system can be found in (Slawik et al., 2014). Overall, the baseline system uses 23 different features. The system is tuned on test2011 and test2012 was used to evaluate the different approaches. In the additional experiments, n-best lists generated for dev2010 and test2010 are used as additional training data for the rescoring.

5.2 Other optimization techniques

For comparison, experimental results include performance obtained with the most widely used algorithms: MERT, KB-MIRA (Cherry and Foster, 2012) as implemented in Moses (Koehn et al., 2007), along with the PRO algorithm. For the latter, we used the MegaM[1] version (Daumé III, 2004). All the results correspond to three random restarts and the weights are chosen according to the best performance on the development data.

5.3 WMT – English to German

The results for the English to German news translation task are summarized in Table 1. The translations generated by the phrase-based decoder reach a BLEU score of 20.19. We compared the presented approach with MERT, KB-MIRA and PRO. KB-MIRA and MERT improve the performance by at most 0.3 BLEU points. In contrast, the PRO technique and the ListNet algorithm presented in this paper improve the translation quality by 0.8 BLEU points to 21 BLEU points.

Using the NCE-based or SOUL-based neural network translation models improve the perfor-

[1] `http://www.umiacs.umd.edu/~hal/megam/`

	Baseline		NCE		SOUL		SOUL+NCE	
System	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Baseline		20.19						
MERT	20.63	20.52	21.24	20.92	21.36	20.84	21.36	20.94
KB-MIRA	20.64	20.38	21.51	20.96	21.65	20.83	21.71	21.06
PRO	20.17	**21.01**	21.04	21.25	21.18	21.31	21.14	21.34
ListNet	19.95	20.98	21.00	**21.51**	21.02	**21.54**	21.14	**21.63**

Table 1: WMT Results for English to German

	Baseline		SDWL		SDWL+RBMTM	
System	Dev	Test	Dev	Test	Dev	Test
Baseline		27.77				
MERT	28.18	27.80	28.24	27.65	28.23	27.64
KB-MIRA	28.23	**28.06**	28.18	28.00	28.00	27.88
PRO	27.38	28.01	27.56	28.14	28.68	28.04
ListNet	28.00	27.87	27.89	**28.18**	27.94	**28.28**

Table 2: WMT Results for German to English

mance up to 21.31 using one of the existing algorithms. Again, the best performance was reached using the PRO algorithm. If we use the ListNet algorithm, we can improve the translation score to 21.54 BLEU points. For this condition, this algorithm outperforms the other by 0.2 BLEU point. When using the two models, the ListNet algorithm achieves an additional gain of 0.1 BLEU point. Moreover, we can observe that MERT and KB-MIRA always yield the best results on the development set, whereas BLEU scores on the test set are lower. The opposite trend is observed with ListNet[2] showing a better generalization power.

In summary, in all conditions, the ListNet algorithm outperforms MERT and KB-MIRA. Only in one condition the PRO algorithm generates translations with a BLEU score as high as the ListNet algorithm. The ListNet algorithm outperforms to the best other algorithms by up to 0.3 BLEU points. The baseline translation is improved by 0.8 BLEU points with only conventional features, and by 1.4 BLEU points when using additional models. Furthmore, as shown by the lower scores on the development data, the ListNet algorithm seems to be less prone to overfitting.

5.4 WMT – German to English

The German to English news translation task results are shown in Table 2. The baseline system yields a BLEU score of 27.77 on the test set. This is slightly outperformed by the ListNet algorithm by 0.1 BLEU point. In this configuration, the KB-MIRA-based rescoring and the PRO algorithm slightly outperform the ListNet algorithm by 0.2 BLEU points. MERT generates a BLEU score worse than the ListNet algorithm. When adding the source discriminative word lexicon (SDWL) only or adding this model and the RBM-based translation model, the ListNet based algorithm outperforms again all other models. While the other algorithms could only gain slightly from these models, the ListNet-based optimization improves the BLEU score up to 28.28 points. This is the best performance reached on this task with a 0.1 BLEU point improvement over other optimization algorithms.

5.5 TED – English to German

In addition to the experiments on the news domain, we performed experiments on the task of translating English TED talks into German. The results of these experiments are summarized in Table 3.

In this task, the MERT algorithm performs better than the KB-MIRA and PRO algorithms and generates translations with a BLEU score of 23.46 points. By optimizing the weights of the log-linear model using the ListNet algorithm, we increased the BLEU score slightly to 23.51 points. But in this condition all optimization could not improve the system over the initial translation, which reaches a BLEU score of 23.67 points.

[2] and with PRO to a lesser extent

	Baseline		extra Dev Data	
System	Dev	Test	Dev	Test
Baseline		23.67		
MERT	27.69	23.46	25.63	23.36
KB-MIRA	27.47	23.19	25.65	23.76
PRO	26.67	23.10	25.00	23.65
ListNet	27.37	**23.51**	25.49	**24.08**

Table 3: TED Results for English to German

In addition to the integration of additional features, the rescoring technique also allows an easy facilitation of additional development data. For this task, additional development data is available. Therefore, we also trained all rescoring algorithms on the concatenation of the original development data and the additional two development sets.

The KB-MIRA and PRO algorithm can facilitate this data and generate translation with a higher BLEU score. In contrast, when using the MERT algorithm, the BLEU score is not improved by the additional data. Therefore, the KB-MIRA algorithm performs better than MERT and PRO and can improve the baseline system by 0.1 BLEU points. With the ListNet algorithm it is possible to select translations with a BLEU score that is 0.6 points better than system trained on the smaller development set. The ListNet rescoring improves the baseline system by 0.4 BLEU points and the best other learning algorithm, KB-MIRA, by 0.3 BLEU points.

5.6 Convergence of the ListNet algorithm

To assess the convergence speed of the ListNet algorithm, the Figure 1 plots the evolution of the BLEU+1 score measured on the development set for the English to German translation task. We can observe a fast convergence along with a satisfactory stability. This is an important characteristic of this algorithm in comparison with the randomness exhibited by some usual tuning algorithm such as MERT.

5.7 Score normalization

On the German to English translation task, we compared the normalization of the features used in the previous experiments with normalizing the final score as described in Section 4.1. We evaluated different target feature ranges between 0.5 and 100. The results for these experiments are summarized in Figure 2.

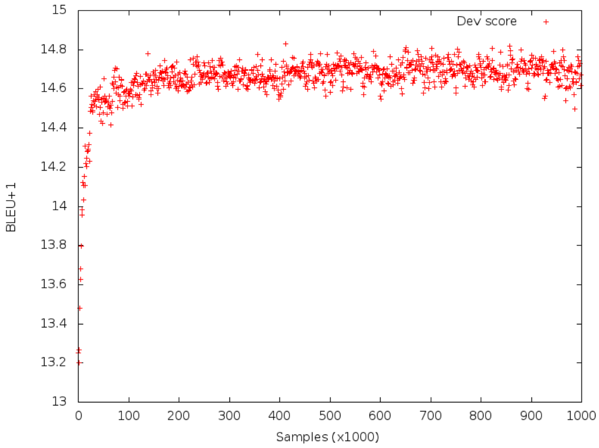

Figure 1: Evolution of the BLEU+1 score measured on the development set as a function of the number of training sentences.

As shown in the graph, if the range of possible scores is too low, no learning is possible. The best performance on the development is reached at a value of ten with 20.21 BLEU points on the development data and 20.64 on the test data. This is also nearly the best performance on the test data.

In comparison, the feature normalization achieves a BLEU score of 19.95 on the development data and 20.98 on the test data as shown in Table 1. Although the normalization of the final score can outperform the feature normalization on the development data, the feature normalization performs best on the test data in this task.

6 Conclusion

We presented in this paper a new way to train the log-linear model of a statistical machine translation system based on an adaptation of the ListNet algorithm to the task of ranking translation hypotheses. This algorithm can be applied to many features and considers the whole n-best list for training. The algorithm can also be applied for

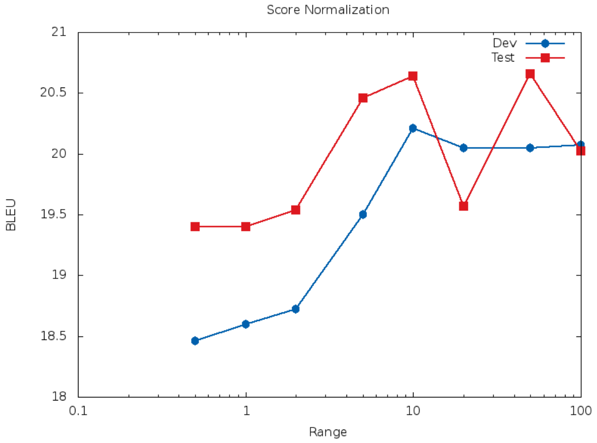

Figure 2: Score normalization

more complex models than the log-linear model used in most machine translation systems.

Using this technique translation quality is improved as measured in BLEU scores on large scale translation tasks. Without any additional feature, we improved the BLEU score by 0.8 points and 0.1 points compared to the initial translations. Further 0.6 BLEU points was gained by using additional models in the rescoring. The algorithm outperformed the MERT training in all configurations and other algorithms in most configurations. Moreover, experimental results show that our approach is less prone to overfitting which is an important issue of many optimization techniques.

Acknowledgments

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452.

References

Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML'07, pages 129–136, New York, NY, USA. ACM.

M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico. 2014. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, California, USA.

W. Chen, T. Liu, Y. Lan, Z. Ma, and H. Li. 2009. Ranking measures and loss functions in learning to rank. In *Advances in Neural Information Processing Systems 22*, pages 315–323.

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, Montréal, Canada, June.

D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, Honolulu, Hawaii, USA.

E. Cho, T. Ha, J. Niehues, T. Herrmann, M. Mediani, Y. Zhang, and A. Waibel. 2015. The karlsruhe institute of technology translation systems for the wmt 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT 2015)*, Lisboa, Portugal.

H. Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression.

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.

X. He and L. 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(ACL 2012)*, pages 292–301, Jeju, Korea.

Teresa Herrmann. 2015. *Linguistic Structure in Statistical Machine Translation*. Ph.D. thesis, Karlsruhe Institute of Technology.

M. Hopkins and J. May. 2011. Tuning as ranking. In *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, Prague, Czech Republic.

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, June. Association for Computational Linguistics.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pages 761–768, Sydney, Australia.

L. Liu, T. Watanabe, E. Sumita, and T. Zhao. 2013. Additive neural networks for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria, Bulgaria.

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)*.

J. Niehues and A. Waibel. 2012. Continuous space language models using restricted boltzmann machines. In *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT 2012)*, Hong Kong.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, and W.-jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318, Philadelphia, Pennsylvania.

A.-V.I. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz. 2011. Expected bleu training for graphs: Bbn system description for wmt11 system combination task. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT 2011)*, pages 159–165, Edinburgh, UK.

I. Slawik, M. Mediani, J. Niehues, Y. Zhang, E. Cho, T. Herrmann, T. Ha, and A. Waibel. 2014. The kit translation systems for iwslt 2014. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, USA.

T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2007)*, Prague, Czech Republic.