The KIT-LIMSI Translation System for WMT 2015

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

Abstract

This paper presented the joined submission of KIT and LIMSI to the English to German translation task of WMT 2015. In this year submission, we integrated a neural network-based translation model into a phrase-based translation model by rescoring the n-best lists.

Since the computation complexity is one of the main issues for continuous space models, we compared two techniques to reduce the computation cost. We investigated models using a structured output layer as well as models trained with noise contrastive estimation. Furthermore, we evaluated a new method to obtain the best log-linear combination in the rescoring phase.

Using these techniques, we were able to improve the BLEU score of the baseline phrase-based system by 1.4 BLEU points.

1 Introduction

In this paper, we present the English→German joint translation system from KIT and LIMSI participating in the Shared Translation Task of the EMNLP 2015 - Tenth Workshop on Statistical Machine Translation (WMT2015). Our system is the combination of two different approaches. First, a strong phrase-based system from KIT is used to generate a k-best list of translated candidates. Second, an n-gram translation model from LIMSI, named *SOUL* (*Structured OUtput Layer*), helps to rescore the k-best list by utilizing features extracted from translated tuples. In this year participation, we also use a version of the neural network translation models (Le et al., 2012) trained using *NCE* algorithm (Gutmann and Hyvärinen, 2010) as counterpart to *SOUL* models. A ListNet-based rescoring method is then applied to integrate two abovementioned approaches.

Section 2 describes the KIT phrase-based translation system which is conducted over the phrase pairs. Section 3 describes the LIMSI *SOUL* and *NCE* translation models estimated on source-and-target n-gram tuples. We explain the rescoring approach in Section 4. Finally, Section 5 summarizes the experimental results of our joint system submitted to WMT2015.

2 KIT Phrase-based Translation System

The KIT translation system uses a phrase-based in-house decoder (Vogel, 2003) which finds the best combinations of features in a log-linear framework. The features consist of translation scores, distortion-based and lexicalized reordering scores as well as conventional and non-word language models. In addition, several reordering rules, including short-range, long-range and tree-based reorderings, are applied before decoding step as they are encoded as word lattices. The decoder then generates a list of the best candidates from the lattices. To optimize the factors of individual features on a development dataset, we use minimum error rate training (MERT) (Venugopal et al., 2005). We are going to describe those components in detail as follows.

2.1 Data and Preprocessing

The parallel data mainly used are the corpora extracted from Europarl Parliament (EPPS), News Commentary (NC) and the common part of web-crawled data (Common Crawl). The monolingual data are the monolingual part of those corpora.

A preprocessing step is applied to the raw data before the actual training. It includes removing excessively long and length-mismatched sentences pairs. Special symbols and nummeric data are normalized, and smartcasing is applied. Sentence pairs which contain textual elements in different

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 120–125,
Lisboa, Portugal, 17-18 September 2015. ©2015 Association for Computational Linguistics.

languages to some extent, are also taken away. The data is further filtered by using an SVM classifier to remove noisy sentences which are not the actual translation from their counterparts.

2.2 Phrase-table Scores

We obtain the word alignments using the GIZA++ toolkit (Och and Ney, 2003) and Discriminative Word Alignment method (Niehues and Vogel, 2008) from the parallel EPPS, NC and Common Crawl. Then the Moses toolkit (Koehn et al., 2007) is used to build the phrase tables. Translation scores, which are used as features in our log-linear framework, are derived from those phrase tables. Additional scores, e.g. distortion information, word penalties and lexicalized reordering probabilities (Koehn et al., 2005), are also extracted from the phrase tables.

2.3 Discriminative Word Lexicon

The presence of words in the source sentence can be used to guide the choice of target words. (Mauser et al., 2009) build a maximum entropy classifier for every target words, taking the presence of source words as its features, in order to predict whether the word should appear in the target sentence or not. In KIT system, we use an extended version described in Niehues and Waibel (2013), which utilizes the presence of source n-grams rather than source words. The parallel data of EPPS and NC are used to train those classifiers.

2.4 Language Models

Besides word-based n-gram language models trained on all preprocessed monolingual data, the KIT system includes several non-word language models. A 4-gram bilingual language model (Niehues et al., 2011) trained on the parallel corpora is used to exploit wider bilingual contexts beyond phrase boundaries. 5-gram Part-of-Speech (POS) language models trained on the POS-tagged parts of all monolingual data incorporate some morphological information into the decision process. They also help to reduce the impact of the data sparsity problem, as cluster language models do. Our 4-gram cluster language model is trained on monolingual EPPS and NC as we use MKCLS algorithm (Och, 1999) to group the words into 1,000 classes and build the language model of the corresponding class IDs instead of the words.

All of the language models are trained using the SRILM toolkit (Stolcke, 2002); The word-based language model scores are estimated by KenLM toolkit (Heafield, 2011) while the non-word language models are estimated by SRILM.

2.5 Prereorderings

The short-range reordering (Rottmann and Vogel, 2007) and long-range reordering (Niehues and Kolss, 2009) rules are extracted from POS-tagged versions of parallel EPPS and NC. The POS tags of those corpora are produced using the TreeTagger (Schmid, 1994). The learnt rules are used to reorder source sentences based on the POS sequences of their target sentences and to build reordering lattices for the translation model. Additionally, a tree-based reordering model (Herrmann et al., 2013) trained on syntactic parse trees (Klein and Manning, 2003) is applied to the source side to better address the differences in word order between English and German.

3 Continuous Space Translation Models

Neural networks, working on top of conventional n-gram back-off language models (BOLMs), have been introduced in (Bengio et al., 2003; Schwenk, 2007) as a potential means to improve discrete language models. More recently, these techniques have been applied to statistical machine translation in order to estimate continuous-space translation models (CTMs) (Schwenk et al., 2007; Le et al., 2012; Devlin et al., 2014)

3.1 n-gram Translation Models

The n-gram-based approach in machine translation is a variant of the phrase-based approach (Koehn et al., 2003). 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 source sentence has been reordered beforehand as illustrated in Figure 1.

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, a matching between a source and target chunks. 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

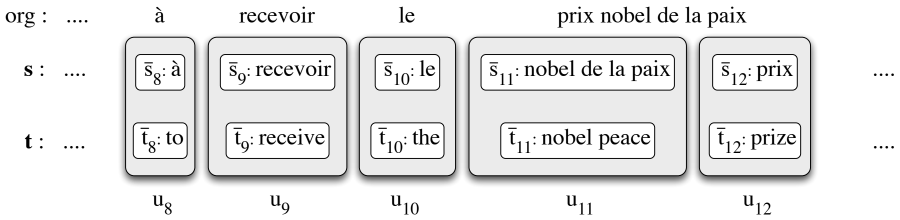

Figure 1: Extract of a French-English sentence pair segmented into bilingual units. The original (*org*) French sentence appears at the top of the figure, just above the reordered source $\mathbf{s}$ and the target $\mathbf{t}$. The pair $(\mathbf{s}, \mathbf{t})$ decomposes into a sequence of L bilingual units (*tuples*) $\mathrm{u}_1, ..., \mathrm{u}_L$. Each tuple u_i contains a source and a target phrase: $\bar{s}_i$ and $\bar{t}_i$.

by-product of source reordering, (see (Crego and Mariño, 2006) for details). During the inference step, the SMT decoder is assumed to output for each source sentence a set of hypotheses along with their derivations, which allow CTMs to score the generated sentence pairs.

Note that the n-gram translation model manipulates bilingual tuples. The underlying set of events is thus much bigger than for word-based models, whereas the training data (parallel corpora) are typically order of magnitude smaller than monolingual resources. As a consequence, data sparsity issues for this model 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) and extended in (Devlin et al., 2014). We assume here the same decomposition.

3.2 Neural Architectures

In such models, the size of output vocabulary is a bottleneck when normalized distributions are needed (Bengio et al., 2003; Schwenk et al., 2007). 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 different alternative, which however only delivers *quasi-normalized* scores, is to train the network using the *Noise Contrastive Estimation* or *NCE* for short (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012). This technique is readily applicable for CTMs. Therefore, *NCE* models deliver a positive score, by applying the exponential function to the output layer activities, instead of the more costly softmax function. We propose here to compare these both approaches, *SOUL* and *NCE* to estimate CTMs. The only difference relies on the output structure of the networks. In terms of computation cost, while the training using the two approaches takes quite similar amounts of time, the inference with *NCE* is slightly faster than the one with *SOUL* as it ignores the score normalization. While the CTMs under study in this paper were initially introduced within the framework of n-gram-based systems (Le et al., 2012), they could be used with any phrase-based system.

Initialization is an important issue when optimizing neural networks. For CTMs, a solution consists in pre-training monolingual n-gram models. Their parameters are then used to initialize bilingual models.

3.3 Integration CTMs

Given the computational cost of computing n-gram probabilities with neural network models, a solution is to resort to a two-pass approach as described in Section 4: the first pass uses a conventional system to produce a k-best list (the k most likely hypotheses); in the second pass, probabilities are computed by the CTMs for each hypothesis and added as new features. Since the phrase-based system described in Section 2 uses source reordering, the decoder was modified to generate k-best lists containing necessary word alignment information between the reordered source sentence and its associated translation. The goal is to recover the information that allows us to apply the n-gram decomposition of a sentence pair.

4 Rescoring

After generating translation probabilities using the neural network translation models, we need to combine them with the baseline scores of the phrase-based system in order to select better translations from the k-best lists. As it is done in the baseline decoder, we used a log-linear combination of all features. We trained the model using the ListNet algorithm (Niehues et al., 2015; Cao et al., 2007).

This technique defines a probability distribution on the permutations of the list based on the scores of the log-linear model and one based on a reference metric. Therefore, a sentence-based translation quality metric is necessary. In our experiments we used the BLEU+1 score introduced by Liang et al. (2006). Then the model was trained by minimizing the cross entropy between both distributions on the development data.

Using this loss function, we can compute the gradient with respect to the weight ω_k as follows:

$$\Delta\omega_k = \sum_{j=1}^{n^{(i)}} f_k(x_j^{(i)}) * \qquad (1)$$

$$\Big(\frac{\exp(f_\omega(x_j^{(i)}))}{\sum_{j'=1}^{n^{(i)}} \exp(f_\omega(x_{j'}^{(i)}))} - \frac{\exp(BLEU(x_j^{(i)}))}{\sum_{j'=1}^{n^i} \exp(BLEU(x_{j'}^{(i)})}\Big)$$

When using the ith sentence, we calculate the derivation by summing over all $n^{(i)}$ items of the k-best lists. The kth feature value $f_k(x_j^{(i)})$ is multiplied with the difference. This difference depends on $f_\omega(x_j^{(i)})$, the score of the log-linear model for the j hypothesis of the list and the BLEU score $BLEU(x_j^{(i)})$ assigned to this item. Using this derivation, we used stochastic gradient descent to train the model. We used batch updates with ten samples and tuned the learning rate on the development data. The training process ends after 100k batches and the final model is selected according to its performance on the development data.

The range of the scores of the different models may greatly differ and many of these values are negative numbers with high absolute value since they are computed as the logarithm of relatively small probabilities. Therefore, we rescale all scores observed on the development data to the range of $[-1, 1]$ prior to reranking.

5 Results

System	Dev	Test
Baseline	20.58	20.19
+ *ListNet* rescoring	19.95	20.98
+ *NCE*	21.00	21.51
+ *SOUL*	21.02	21.54
+ *NCE* + *SOUL*	**21.14**	**21.63**

Table 1: Results of English→German joint system

In this section we present the experimental results of the joint system we submitted for the English→German Shared Translation Task for WMT2015. The systems are tuned on *newtest2013* (Dev) and the BLEU scores we get when applying them over *newtest2014* (Test) are reported in Table 1.

KIT phrase-based system, labeled as the Baseline, reaches 20.58 and 20.19 BLEU points on Dev and Test sets, respectively. Using our new rescoring ListNet-based instead of traditional MERT yields upto 0.8 BLEU points. Adding features estimated from different neural architectures of CTMs gains a further 0.56 BLEU point improvement. More precisely, when CTMs scores are computed using neural networks trained with *NCE* output layer and added to the new k-best list for rescoring, we can observe that the BLEU score on the test set achieves 21.51. With similar procedures using *SOUL* output layer, the gain is slightly better, reaching 21.54. Finally, adding all of the scores derived from those two alternative output structures results to our submitted system with the BLEU of 21.63, which is 1.4 BLEU points different from the baseline system.

Expensive computational cost is an important issue while using CTMs estimated on large vocabularies (Section 3.2). Table 2 compares the training and inference speed for *SOUL* and *NCE* models. While the two kinds of models have a same speed in training, in inference the *NCE* models benefit from their un-normalized scoring. Both ap-

	training speed	inference speed
SOUL	1000 / s	15500 / s
NCE	1000 / s	19400 / s

Table 2: Speeds of the training and the inference corresponding to *SOUL* and *NCE* models, expressed in number of processed words per second.

proaches are plausible workarounds to overcome the computational difficulty by speeding up both the training and the inference, contrary to some propositions in the literature which only reduces the inference time (Devlin et al., 2014).

6 Conclusion

In the experiments we showed that a strong baseline phrase-based translation system, which already used several models during decoding, could be improved significantly by adding computational complex models in a rescoring step.

Firstly, in our experiments, the translation quality was improved by rescoring the n-best list of the baseline system. We could improve the BLEU score by 0.8 points without adding additional features. When adding CTMs features, additional gains of 0.6 BLEU points were achieved.

Secondly, we compared two approaches to limit the computation complexity of continuous space models. The *SOUL* and *NCE* models perform similarly; both improved the translation quality by 0.5 points. Small additional gains of 0.1 BLEU points were achieved by using both models together.

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

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

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: from Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 129–136. ACM.

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

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

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, Maryland.

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.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2013. Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, Georgia, USA, June. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of ACL 2003*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *HLT/NAACL 2003*.

Philipp Koehn, Amittai Axelrod, Alexandra B. Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, Demonstration Session*, 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 ICASSP*, 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.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An End-to-end Discriminative Approach to Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 761–768. Association for Computational Linguistics.

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.

Arne Mauser, Saša Hasan, and Hermann Ney. 2009. Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, EMNLP '09, Singapore.

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 Muntsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece.

Jan Niehues and Stephan Vogel. 2008. Discriminative Word Alignment via Alignment Matrix Modeling. In *Proceedings of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA.

Jan Niehues and Alex Waibel. 2013. An MT Error-driven Discriminative Word Lexicon Using Sentence Structure Features. In *Proceedings of the Eighth Workshop on Statistical Machine Translation, Sofia, Bulgaria*, pages 512–520.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider Context by Using Bilingual Language Models in Machine Translation. In *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK.

Jan Niehues, Quoc Khanh Do, Alexandre Allauzen, and Alex Waibel. 2015. ListNet-based MT Rescoring. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT 2015)*, Lisboa, Portugal.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 1999. An Efficient Method for Determining Bilingual Word Classes. In *EACL'99*.

Kay Rottmann and Stephan Vogel. 2007. Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, United Kingdom.

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, July.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *International Conference on Spoken Language Processing*, Denver, Colorado, USA.

Ashish Venugopal, Andreas Zollman, and Alex Waibel. 2005. Training and Evaluating Error Minimization Rules for Statistical Machine Translation. In *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, Michigan, USA.

Stephan Vogel. 2003. SMT Decoder Dissected: Word Reordering. In *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China.