The KIT-LIMSI Translation System for WMT 2014

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

Abstract

This paper describes the joined submission of LIMSI and KIT to the Shared Translation Task for the German-to-English direction. The system consists of a phrase-based translation system using a pre-reordering approach. The baseline system already includes several models like conventional language models on different word factors and a discriminative word lexicon. This system is used to generate a k-best list. In a second step, the list is reranked using *SOUL* language and translation models (Le et al., 2011).

Originally, *SOUL* translation models were applied to n-gram-based translation systems that use tuples as translation units instead of phrase pairs. In this article, we describe their integration into the KIT phrase-based system. Experimental results show that their use can yield significant improvements in terms of BLEU score.

1 Introduction

This paper describes the KIT-LIMSI system for the Shared Task of the ACL 2014 Ninth Workshop on Statistical Machine Translation. The system participates in the German-to-English translation task. It consists of two main components. First, a k-best list is generated using a phrase-based machine translation system. This system will be described in Section 2. Afterwards, the k-best list is reranked using *SOUL* (*Structured OUtput Layer*) models. Thereby, a neural network language model (Le et al., 2011), as well as several translation models (Le et al., 2012a) are used. A detailed description of these models can be found in Section 3. While the translation system uses phrase pairs, the *SOUL* translation model uses tuples as described in the n-gram approach (Mariño et al., 2006). We describe the integration of the *SOUL* models into the translation system in Section 3.2. Section 4 summarizes the experimental results and compares two different tuning algorithms: Minimum Error Rate Training (Och, 2003) and k-best Batch Margin Infused Relaxed Algorithm (Cherry and Foster, 2012).

2 Baseline system

The KIT translation system is an in-house implementation of the phrase-based approach and includes a pre-ordering step. This system is fully described in Vogel (2003).

To train translation models, the provided Europarl, NC and Common Crawl parallel corpora are used. The target side of those parallel corpora, the News Shuffle corpus and the GigaWord corpus are used as monolingual training data for the different language models. Optimization is done with Minimum Error Rate Training as described in Venugopal et al. (2005), using newstest2012 and newstest2013 as development and test data, respectively.

Compound splitting (Koehn and Knight, 2003) is performed on the source side (German) of the corpus before training. Since the web-crawled Common Crawl corpus is noisy, this corpus is first filtered using an SVM classifier as described in Mediani et al. (2011).

The word alignment is generated using the GIZA++ Toolkit (Och and Ney, 2003). Phrase extraction and scoring is done using the Moses toolkit (Koehn et al., 2007). Phrase pair probabilities are computed using modified Kneser-Ney smoothing (Foster et al., 2006).

We apply short-range reorderings (Rottmann and Vogel, 2007) and long-range reorderings (Niehues and Kolss, 2009) based on part-of-speech tags. The POS tags are generated using the TreeTagger (Schmid, 1994). Rewriting rules

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 84–89,
Baltimore, Maryland USA, June 26–27, 2014.

based on POS sequences are learnt automatically to perform source sentence reordering according to the target language word order. The long-range reordering rules are further applied to the training corpus to create reordering lattices to extract the phrases for the translation model. In addition, a tree-based reordering model (Herrmann et al., 2013) trained on syntactic parse trees (Rafferty and Manning, 2008; Klein and Manning, 2003) is applied to the source sentence. In addition to these pre-reordering models, a lexicalized reordering model (Koehn et al., 2005) is applied during decoding.

Language models are trained with the SRILM toolkit (Stolcke, 2002) using modified Kneser-Ney smoothing (Chen and Goodman, 1996). The system uses a 4-gram word-based language model trained on all monolingual data and an additional language model trained on automatically selected data (Moore and Lewis, 2010). The system further applies a language model based on 1000 automatically learned word classes using the MKCLS algorithm (Och, 1999). In addition, a bilingual language model (Niehues et al., 2011) is used as well as a discriminative word lexicon (DWL) using source context to guide the word choices in the target sentence.

3 *SOUL* models for statistical machine translation

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. The *SOUL* model (Le et al., 2011) is a specific neural network architecture that allows us to estimate n-gram models using large vocabularies, thereby making the training of large neural network models feasible both for target language models and translation models (Le et al., 2012a).

3.1 *SOUL* translation models

While the integration of *SOUL* target language models is straightforward, *SOUL* translation models rely on a specific decomposition of the joint probability $P(\mathbf{s}, \mathbf{t})$ of a sentence pair, where $\mathbf{s}$ is a sequence of I *reordered* source words $(s_1, ..., s_I)$[1] and $\mathbf{t}$ contains J target words $(t_1, ..., t_J)$. In the n-gram approach (Mariño et al., 2006; Crego et al., 2011), this segmentation is a by-product of source reordering, and ultimately derives from initial word and phrase alignments. In this framework, the basic translation units are *tuples*, which are analogous to phrase pairs, and represent a matching $\mathrm{u} = (\overline{s}, \overline{t})$ between a source phrase $\overline{s}$ and a target phrase $\overline{t}$.

Using the n-gram assumption, the joint probability of a segmented sentence pair using L tupels decomposes as:

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} P(\mathrm{u}_i | \mathrm{u}_{i-1}, ..., \mathrm{u}_{i-n+1}) \quad (1)$$

A first issue with this decomposition is that the elementary units are bilingual pairs. Therefore, the underlying vocabulary and hence the number of parameters can be quite large, even for small translation tasks. Due to data sparsity issues, such models are bound to face severe estimation problems. Another problem with Equation (1) is that the source and target sides play symmetric roles, whereas the source side is known, and the target side must be predicted. To overcome some of these issues, the n-gram probability in Equation (1) can be factored by first decomposing tuples in two (source and target) parts, and then decomposing the source and target parts at the word level.

Let s_i^k denote the k^{th} word of source part of the tuple $\overline{s}_i$. Let us consider the example of Figure 1, s_{11}^1 corresponds to the source word *nobel*, s_{11}^4 to the source word *paix*, and similarly t_{11}^2 is the target word *peace*. We finally define $h^{n-1}(t_i^k)$ as the sequence of the $n-1$ words preceding t_i^k in the target sentence, and $h^{n-1}(s_i^k)$ as the $n-1$ words preceding s_i^k in the reordered source sentence: in Figure 1, $h^3(t_{11}^2)$ thus refers to the three word context *receive the nobel* associated with the target word *peace*. Using these notations, Equation 1 can be rewritten as:

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} \Big[\prod_{k=1}^{|\overline{t}_i|} P\big(t_i^k | h^{n-1}(t_i^k), h^{n-1}(s_{i+1}^1)\big) \times \prod_{k=1}^{|\overline{s}_i|} P\big(s_i^k | h^{n-1}(t_i^1), h^{n-1}(s_i^k)\big) \Big] \quad (2)$$

This decomposition relies on the n-gram assumption, this time at the word level. Therefore, this

[1] In the context of the n-gram translation model, $(\mathbf{s}, \mathbf{t})$ thus denotes an *aligned* sentence pair, where the source words are reordered.

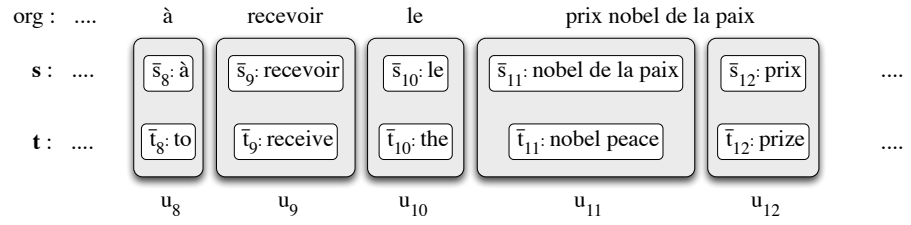

Figure 1: Extract of a French-English sentence pair segmented into bilingual units. The original (*org*) French sentence appears at the top of the figure, just above the reordered source $\mathbf{s}$ and the target $\mathbf{t}$. The pair $(\mathbf{s}, \mathbf{t})$ decomposes into a sequence of L bilingual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

model estimates the joint probability of a sentence pair using two sliding windows of length n, one for each language; however, the moves of these windows remain synchronized by the tuple segmentation. Moreover, the context is not limited to the current phrase, and continues to include words in adjacent phrases. Equation (2) involves two terms that will be further denoted as *TrgSrc* and *Src*, respectively $P\big(t_i^k|h^{n-1}(t_i^k), h^{n-1}(s_{i+1}^1)\big)$ and $P\big(s_i^k|h^{n-1}(t_i^1), h^{n-1}(s_i^k)\big)$. It is worth noticing that the joint probability of a sentence pair can also be decomposed by considering the following two terms: $P\big(s_i^k|h^{n-1}(s_i^k), h^{n-1}(t_{i+1}^1)\big)$ and $P\big(t_i^k|h^{n-1}(s_i^1), h^{n-1}(t_i^k)\big)$. These two terms will be further denoted by *SrcTrg* and *Trg*. Therefore, adding *SOUL* translation models means that 4 scores are added to the phrase-based systems.

3.2 Integration

During the training step, the *SOUL* translation models are trained as described in (Le et al., 2012a). The main changes concern the inference step. Given the computational cost of computing n-gram probabilities with neural network models, a solution is to resort to a two-pass approach: 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 *SOUL* models for each hypothesis and added as new features. Then the k-best list is reordered according to a combination of all features including these new features. In the following experiments, we use 10-gram *SOUL* models to rescore 300-best lists. Since the phrase-based system described in Section 2 uses source reordering, the decoder was modified in order to generate k-best lists that contain necessary word alignment information between the reordered source sentence and its associated target hypothesis. The goal is to recover the information that is illustrated in Figure 1 and to apply the n-gram decomposition of a sentence pair.

These (target and bilingual) neural network models produce scores for each hypothesis in the k-best list; these new features, along with the features from the baseline system, are then provided to a new phase which runs the traditional Minimum Error Rate Training (*MERT*) (Och, 2003), or a recently proposed k-best Batch Margin Infused Relaxed Algorithm (*KBMIRA*) (Cherry and Foster, 2012) for tuning purpose. The *SOUL* models used for this year's evaluation are similar to those described in Allauzen et al. (2013) and Le et al. (2012b). However, since compared to these evaluations less parallel data is available for the German-to-English task, we use smaller vocabularies of about $100K$ words.

4 Results

We evaluated the SOUL models on the German-to-English translation task using two systems to generate the k-best lists. The first system used all models of the baseline system except the DWL model and the other one used all models.

Table 1 summarizes experimental results in terms of BLEU scores when the tuning is performed using *KBMIRA*. As described in Section 3, the probability of a phrase pair can be decomposed into products of words' probabilities in 2 different ways: we can first estimate the probability of words in the source phrase given the context, and then the probability of the target phrase given its associated source phrase and context words (see Equation (2)); or inversely we can generate the target side before the source side. The former proceeds by adding *Src* and *TrgSrc* scores as

	No DWL		DWL	
Soul models	Dev	Test	Dev	Test
No	26.02	27.02	26.27	27.46
Target	26.30	27.42	26.43	27.85
Translation st	26.46	27.70	26.66	28.04
Translation ts	26.48	27.41	26.61	28.00
All Translation	26.50	27.86	26.70	28.08
All *SOUL* models	26.62	27.84	26.75	28.10

Table 1: Results using *KBMIRA*

	No DWL		DWL	
Soul models	Dev	Test	Dev	Test
No	26.02	27.02	26.27	27.46
Target	26.18	27.09	26.44	27.54
Translation st	26.36	27.59	26.66	27.80
Translation ts	26.44	27.69	26.63	27.94
All Translation	26.53	27.65	26.69	27.99
All *SOUL* models	26.47	27.68	**26.66**	**28.01**

Table 2: Results using *MERT*. Results in bold correpond to the submitted system.

2 new features into the k-best list, and the latter by adding *Trg* and *SrcTrg* scores. These 2 methods correspond respectively to the *Translation ts* and *Translation st* lines in the Table 1. The 4 translation models may also be added simultaneously (*All Translations*). The first line gives baseline results without *SOUL* models, while the *Target* line shows results in adding only *SOUL* language model. The last line (*All SOUL models*) shows the results for adding all neural network models into the baseline systems.

As evident in Table 1, using the *SOUL* translation models yields generally better results than using the *SOUL* target language model, yielding about 0.2 BLEU point differences on dev and test sets. We can therefore assume that the *SOUL* translation models provide richer information that, to some extent, covers that contained in the neural network language model. Indeed, these 4 translation models take into account not only lexical probabilities of translating target words given source words (or in the inverse order), but also the probabilities of generating words in the target side (*Trg* model) as does a language model, with the same context length over both source and target sides. It is therefore not surprising that adding the *SOUL* language model along with all translation models (the last line in the table) does not give significant improvement compared to the other configurations. The different ways of using the *SOUL* translation models perform very similarly.

Table 2 summarizes the results using *MERT* instead of *KBMIRA*. We can observe that using *KBMIRA* results in 0.1 to 0.2 BLEU point improvements compared to *MERT*. Moreover, this impact becomes more important when more features are considered (the last line when all 5 neural network models are added into the baseline systems). In short, the use of neural network models yields up to 0.6 BLEU improvement on the DWL system, and a 0.8 BLEU gain on the system without DWL. Unfortunately, the experiments with *KBMIRA* were carried out after the the submission date. Therefore the submitted system corresponds to the last line of table 2 indicated in bold.

5 Conclusion

We presented a system with two main features: a phrase-based translation system which uses prereordering and the integration of *SOUL* target language and translation models. Although the translation performance of the baseline system is already very competitive, the rescoring by *SOUL* models improve the performance significantly. In the rescoring step, we used a continuous language model as well as four continuous translation mod-

els. When combining the different *SOUL* models, the translation models are observed to be more important in increasing the translation performance than the language model. Moreover, we observe a slight benefit to use KBMIRA instead of the standard MERT tuning algorithm. It is worth noticing that using KBMIRA improves the performance but also reduces the variance of the final results.

As future work, the integration of the *SOUL* translation models could be improved in different ways. For *SOUL* translation models, there is a mismatch between translation units used during the training step and those used by the decoder. The former are derived using the n-gram-based approach, while the latter use the conventional phrase extraction heuristic. We assume that reducing this mismatch could improve the overall performance. This can be achieved for instance using forced decoding to infer a segmentation of the training data into translation units. Then the *SOUL* translation models can be trained using this segmentation. For the *SOUL* target language model, in these experiments we only used the English part of the parallel data for training. Results may be improved by including all the monolingual data.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658 as well as the French Armaments Procurement Agency (DGA) under the RAPID Rapmat project.

References

Alexandre Allauzen, Nicolas Pécheux, Quoc Khanh Do, Marco Dinarelli, Thomas Lavergne, Aurélien Max, Hai-Son Le, and François Yvon. 2013. Limsi@ wmt13. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 60–67.

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

S.F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics (ACL '96)*, pages 310–318, Santa Cruz, California, USA.

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

Josep M. Crego, Franois Yvon, and Jos B. Mariño. 2011. N-code: an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

George F. Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *EMNLP*, pages 53–61.

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 and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL*, Budapest, Hungary.

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. 2012a. Continuous space translation models with neural networks. pages 39–48, Montréal, Canada, June. Association for Computational Linguistics.

Hai-Son Le, Thomas Lavergne, Alexandre Allauzen, Marianna Apidianaki, Li Gong, Aurélien Max, Artem Sokolov, Guillaume Wisniewski, and François Yvon. 2012b. Limsi@ wmt'12. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 330–337. 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.

Mohammed Mediani, Eunah Cho, Jan Niehues, Teresa Herrmann, and Alex Waibel. 2011. The KIT English-French Translation systems for IWSLT 2011. In *Proceedings of the Eight International Workshop on Spoken Language Translation (IWSLT)*.

R.C. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

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, 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.

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*.

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

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the Workshop on Parsing German*.

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. 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.