Using Factored Word Representation in Neural Network Language Models

Jan Niehues, Thanh-Le Ha, Eunah Cho and Alex Waibel
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.secondname@kit.edu

Abstract

Neural network language and translation models have recently shown their great potentials in improving the performance of phrase-based machine translation. At the same time, word representations using different word factors have been translation quality and are part of many state-of-the-art machine translation systems. used in many state-of-the-art machine translation systems, in order to support better translation quality.

In this work, we combined these two ideas by investigating the combination of both techniques. By representing words in neural network language models using different factors, we were able to improve the models themselves as well as their impact on the overall machine translation performance. This is especially helpful for morphologically rich languages due to their large vocabulary size. Furthermore, it is easy to add additional knowledge, such as source side information, to the model.

Using this model we improved the translation quality of a state-of-the-art phrase-based machine translation system by 0.7 BLEU points. We performed experiments on three language pairs for the news translation task of the WMT 2016 evaluation.

1 Introduction

Recently, neural network models are deployed extensively for better translation quality of statistical machine translation (Le et al., 2011; Devlin et al., 2014). For the language model as well as for the translation model, neural network-based models showed improvements when used during decoding as well as when used in re-scoring.

In phrase-based machine translation (PBMT), word representation using different factors (Koehn and Hoang, 2007) are commonly used in state-of-the-art systems. Using Part-of-Speech (POS) information or automatic word clusters is especially important for morphologically rich languages which often have a large vocabulary size. Language models based on these factors are able to consider longer context and therefore improve the modelling of the overall structure. Furthermore, the POS information can be used to improve the modelling of word agreement, which is often a difficult task when handling morphologically rich languages.

Until now, word factors have been used relatively limited in neural network models. Automatic word classes have been used to structure the output layer (Le et al., 2011) and as input in feed forward neural network language models (Niehues and Waibel, 2012).

In this work, we propose a multi-factor recurrent neural network (RNN)-based language model that is able to facilitate all available information about the word in the input as well as in the output. We evaluated the technique using the surface form, POS-tag and automatic word clusters using different cluster sizes.

Using this model, it is also possible to integrate source side information into the model. By using the model as a bilingual model, the probability of the translation can be modelled and not only the one of target sentence. As for the target side, we use a factored representation for the words on the source side.

The remaining of the paper is structured as following: In the following section, we first review the related work. Afterwards, we will shortly describe the RNN-based language model used in our experiments. In Section 4, we will introduce the factored RNN-based language model. In the next

Proceedings of the First Conference on Machine Translation, Volume 1: Research Papers, pages 74–82,
Berlin, Germany, August 11-12, 2016.

section, we will describe the experiments on the WMT 2016 data. Finally, we will end the paper with a conclusion of the work.

2 Related Work

Additional information about words, encoded as word factors, e.g. the lemma of word, POS tags, etc., is employed in state-of-the-art phrase-based systems. (Koehn and Hoang, 2007) decomposes the translation of factored representations to smaller mapping steps, which are modelled by translation probabilities from input factor to output factor or by generating probabilities of additional output factors from existing output factors. Then those pre-computed probabilities are jointly combined in the decoding process as a standard translation feature scores. In addition, language models using these word factors have shown to be very helpful to improve the translation quality. In particular, the aligned-words, POS or word classes are used in the framework of modern language models (Mediani et al., 2011; Wuebker et al., 2013).

Recently, neural network language models have been considered to perform better than standard n-gram language models (Schwenk, 2007; Le et al., 2011). Especially the neural language models constructed in recurrent architectures have shown a great performance by allowing them to take a longer context into account (Mikolov et al., 2010; Sundermeyer et al., 2013).

In a different direction, there has been a great deal of research on bringing not only target words but also source words into the prediction process, instead of predicting the next target word based on the previous target words (Le et al., 2012; Devlin et al., 2014; Ha et al., 2014).

However, to the best of our knowledge, word factors have been exploited in a relatively limited scope of neural network research. (Le et al., 2011; Le et al., 2012) use word classes to reduce the output layer's complexity of such networks, both in language and translation models. In the work of (Niehues and Waibel, 2012), their Restricted Boltzmann Machines language models also encode word classes as an additional input feature in predicting the next target word. (Tran et al., 2014) use two separate feed forward networks to predict the target word and its corresponding suffixes with the source words and target stem as input features.

Our work exhibits several essential differences from theirs. Firstly, we leverage not only the target morphological information but also word factors from both source and target sides in our models. Furthermore, we could use as many types of word factors as we can provide. Thus, we are able to make the most of the information encoded in those factors for more accurate prediction.

3 Recurrent Neural Network-based Language Models

In contrast to feed forward neural network-based language models, recurrent neural network-based language models are able to store arbitrary long word sequences. Thereby, they are able to directly model $P(w|h)$ and no approximations by limiting the history size are necessary. Recently, several authors showed that RNN-based language models could perform very well in phrase-based machine translation. (Mikolov et al., 2010; Sundermeyer et al., 2013)

In this work, we used the torch7[1] implementation of an RNN-based language model (Léonard et al., 2015). First, the words were mapped to their word embeddings. We used an input embedding size of 100. Afterwards, we used two LSTM-based layers. The first has the size of the word embeddings and for the second we used a hidden size of 200. Finally, the word probabilities were calculated using a softmax layer.

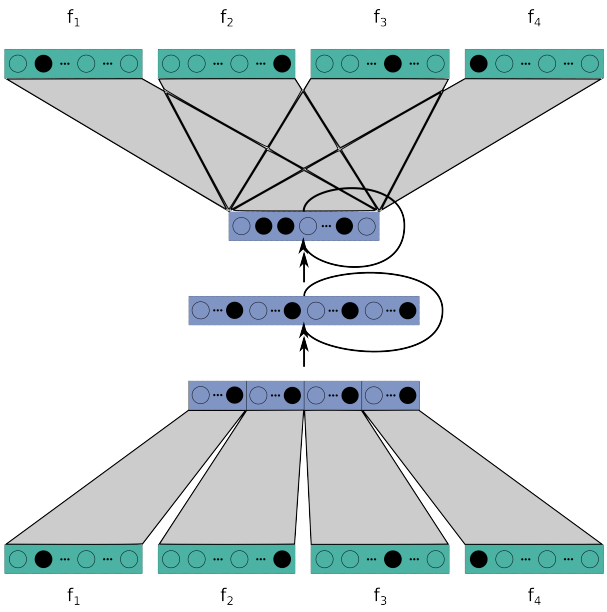
The models were trained using stochastic gradient descent. The weights were updated using mini-batches with a batch size of 128. We used a maximum epoch size of 1 million examples and selected the model with the lowest perplexity on the development data.

4 Factored Language Model

When using factored representation of words, words are no longer represented as indices in the neural network. Instead, they are represented a tuples of indices $w = (f_1, \ldots, f_D)$, where D is the number of different factors used to describe the word. These factors can be the word itself, as well as the POS, automatic learned classes (Och, 1999) or other information about the word. Furthermore, we can use different types of factors for the input and the output of the neural network.

[1] http://torch.ch/

Figure 1: Factored RNN Layout



4.1 Input Representation

In a first step, we obtained a factored representation for the input of the neural network. In the experiments, we represented a word by its surface form, POS-tags and automatic word class, but the framework can be used for any number of word factors. Although there are factored approaches for n-gram based language models (Bilmes and Kirchhoff, 2003), most n-gram language models only use one factor. In contrast, in neural network based language models, it is very easy to add additional information as word factors. We can learn different embeddings for each factor and represent the word by concatenating the embeddings of several factors. As shown in the bottom of Figure 1, we first project the different factors to the continuous factor embeddings. Afterwards, we concatenate these embeddings into a word embedding.

The advantage of using several word factors is that we can use different knowledge sources to represent a word. When a word occurs very rarely, the learned embedding from its surface form might not be helpful. The additional POS information, however, is very helpful. While using POS-based language models in PBMT may lead to losing the information about high frequent words, in this approach we can have access to all information by concatenating the factor embeddings.

4.2 Output Representation

In addition to use different factors in the input of the neural network, we can also use different factors on the output. In phrase-based machine translation, n-gram language models based on POS-tags have been shown to be very successful for morphologically rich languages.

Porting this idea to neural network language models, we can not only train a model to predict the original word f_1 given the previous words in factor representation $h = (f_{1,1}, \ldots, f_{1,D}), \ldots, (f_{i,1}, \ldots, f_{i,D})$, but also train a model to predict he POS-tags (e.g. f_2) given the history h.

In a first step, we proposed to train individual models for all factors $1, \ldots, D$ generating probabilities $P_1, \ldots, P_D$ for every sentence. Then these probabilities can be used as features for example in re-scoring of the phrase-based MT system.

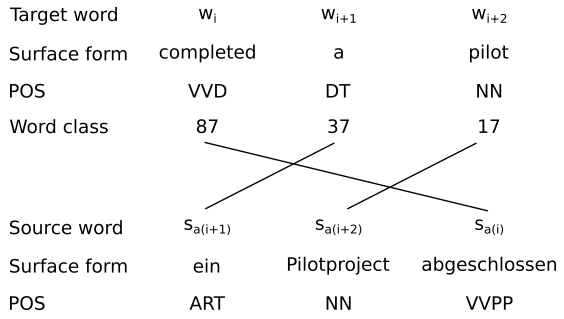
Considering that it can be helpful to consider all factors of the word in the input, it can be also helpful to jointly train the models for predicting the different output factors. This is motivated by the fact that multi-task learning has shown to be beneficial in several NLP tasks (Collobert et al., 2011). Predicting all output features jointly requires a modification of the output layer of the RNN model. As shown in Figure 1, we replace the single mapping from the LSTM-layer to the softmax layer, by D mappings. Each mapping then learns to project the LSTM-layer output to the factored output probabilities. In the last layer, we use D different softmax units. In a similar way as the conventional network, the error between the output of the network and the reference is calculated during training.

Using this network, we will no longer predict the probability of one word factor $P_d, d \in \{1, \ldots D\}$, but D different probability distributions $P_1, \ldots, P_D$. In order to integrate this model into the machine translation system we explored two different probabilities. First, we used only the joint probability $P = \prod_{d=1}^{D} P_d$ as a feature in the log-linear combination. In addition, we also used the joint probability as well as all individual probabilities P_d as features.

4.3 Bilingual Model

Using the model presented before, it is possible to add additional information to the model as well. One example we explored in this work is to use

Figure 2: Bilingual Model



the model as a bilingual model (BM). Instead of using only monolingual information by considering the previous target factors as input, we used source factors additionally. Thereby, we can now model the probability of a word given the previous target words and information about the source sentence. So in this case we model the translation probability and no longer the language model probability.

When predicting the target word w_{i+1} with its factors $f_{i+1,1}, \ldots, f_{i+1,D}$, the input to the RNN is the previous target word $w_i = f_{i,1}, \ldots, f_{i,D}$. Using the alignment, we can find the source word $s_{a(i+1)}$, which is aligned to the target word w_{i+1}. When we add the features of source word

$$s_{a(i+1)} = (f^s_{a(i+1),1}, \ldots, f^s_{a(i+1),D_s})$$

to the ones of the target word w_i and create a new bilingual token

$$b_i = (f_{i+1,1}, \ldots, f_{i+1,D}, f^s_{a(i+1),1}, \ldots, f^s_{a(i+1),D_s})$$

, we now can predict the target word given the previous target word and the aligned source word.

In the example in Figure 2, we would insert (completed,VVD,87,ein,ART) to predict (a,DT,37).

In this case the number of input factors and output factors are no longer the same. In the input, we have $D + D_s$ input factors, while we have only D factors on the output of the network.

5 Experiments

We evaluated the factored RNNLM on three different language pairs of the WMT 2016 News Translation Task. In each language pair, we created an n-best list using our phrase-based MT system and used the factored RNNLM as an additional feature in rescoring. It is worth noting that the POS and word class information are already present during decoding of the baseline system by n-gram-based language models based on each of these factors. First, we performed a detailed analysis on the English-Romanian task. In addition, we used the model in a German-English and English-German translation system. In all tasks, we used the model in re-scoring of a PBMT system.

5.1 System Description

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 (Rottmann and Vogel, 2007; Niehues and Kolss, 2009; Herrmann et al., 2013) and facilitates several translation and language models. As shown in Table 1, we use two to three word-based language models and one to two cluster-based models using 50, 100 or 1,000 clusters. The custers were trained as described in (Och, 1999). In addition, we used a POS-based language model in the English-Romainian system and a bilingual language model (Niehues et al., 2011) in English to German and German to English systems. The POS tags for English-Romanian were generated by the tagger described in (Ion et al., 2012) and the ones for German by RFTagger (Schmid and Laws, 2008).

Table 1: *Features*

	EN-RO	EN-DE	DE-EN
wordLM	2	3	3
POSLM	1	0	0
clusterLM	2	1	2
BiLM	0	1	1
#features	22-23	20	22

In addition, we used discriminative word lexica (Niehues and Waibel, 2013) during decoding and source discriminative word lexica in rescoring (Herrman et al., 2015).

A full system description can be found in (Ha et al., 2016).

The German to English baseline system uses 20 features and the English to German systems uses 22 features.

The English-Romanian system was optimized on the first part of news-dev2016 and the rescoring was optimized on this set and a subset of 2,000

sentences from the SETimes corpus. This part of the corpus was of course excluded for training the model. The system was tested on the second half of news-dev2016.

The English-German and German-English systems were optimized on news-test2014 and also the re-scoring was optimized on this data. We tested the system on news-test2015.

For English to Romanian and English to German we used an n-best List of 300 entries and for German to English we used an n-best list with 3,000 entries.

For decoding, for all language directions, the weights of the system were optimized using minimum error rate training (Och, 2003). The weights in the rescoring were optimized using the ListNet algorithm (Cao et al., 2007) as described in (Niehues et al., 2015).

The RNN-based language models for English to Romanian and German to English were trained on the target side of the parallel training data. For English to German, we trained the model and the Europarl corpus and the News commentary corpus.

5.2 English - Romanian

In the first experiment on the English to Romanian task, we only used the scores of the RNN language models. The baseline system has a BLEU score (Papineni et al., 2002) of 29.67. Using only the language model instead of the 22 features, of course, leads to a lower performance, but we can see clear difference between the different language models. All systems use a word vocabulary of 5K words and we used four different factors. We used the word surface form, the POS tags and word clusters using 100 and 1,000 classes.

The baseline model using words as input and words as output reaches a BLEU score of 27.88. If we instead represent the input words by factors, we select entries from the n-best list that generates a BLEU score of 28.46. As done with the n-gram language models, we can also predict the other factors instead of the words themselves. In all cases, we use all four factors as input factors. As shown in Table 2, all models except for the one with 100 classes perform similarly, reaching up between 28.46 and 28.49. The language model predicting only 100 classes only reaches a BLEU score of 28.23. It suggests that this number of classes is too low to disambiguate the entries in the n-best list.

Table 2: *English - Romanian Single Score*

Input	Prediction	Single
Word	Word	27.88
All factors	Word	28.46
All factors	POS	28.48
All factors	100 Cl.	28.23
All factors	1,000 Cl.	28.49
All factors	All factors	28.54

If we predict all factors together and use then the joint probability, we can reach the best BLEU score of 28.54 as shown in the last line of the table. This is 0.7 BLEU points better than the initial word based model.

After evaluating the model as the only knowledge source, we also performed experiments using the model in combination with the other models. We evaluated the baseline and the best model in three different configuration in Table 3 using only the joint probability. The three baseline configuration differ in the models used during decoding. Thereby, we are able to generate different n-best lists and test the models on different conditions.

Table 3: *English - Romanian Language Models*

Model	Conf1	Conf2	Conf3
Baseline	29.86	30.00	29.75
LM 5K	29.79	29.84	29.73
LM 50K	29.64	29.84	29.83
Factored LM 5K	29.94	30.01	30.01
Factored LM 50K	30.05	30.27	30.29

In Table 3, we tested the word-based and the factored language model using a vocabulary of 5K and 50K words. Features from each model are used in addition to the features of the baseline system. As shown in the table, the word-based RNN language models perform similarly, but both could not improve over the baseline system. One possible reason for this is that we already use several language models in the baseline model and they are partly trained on much larger data. While the RNN models are trained using only the target language model, one word-based language model is trained on the Romanian common crawl corpus. Furthermore, the POS-based and word cluster language models use a 9-gram history and therefore, can already model quite long dependencies.

But if we use a factored language model, we are

able to improve over the baseline system. Using the additional information of the other word factors, we are able to improve the bilingual model in all situations. The model using a surface word vocabulary of 5,000 words can improve by 0.1 to 0.3 BLEU points. The model using a 50K vocabulary can even improve by up to 0.6 BLEU points.

Table 4: *English - Romanian Bilingual Models*

Model	Dev	Test
Baseline	40.12	29.75
+ Factored LM 50K	40.87	30.17
+ Factored BM 5K	41.11	30.44
+ Factored BM 50K	41.16	30.57

After analyzing the different language models, we also evaluate how we can use the factored representation to include source side information. The results are summarized in Table 4. In these experiments, we used not only the the joint probability, but also the four individual probabilities as features. Therefore, we will add five scores for every model, since each model is added to its previous configuration in this experiment.

Exploiting all five probabilities of the language model brought us the similar improvement we achieved using the joint probability from the model. On the test set, the improvements are slightly worse. When adding the model using source side information based on a vocabulary of 5K and 50K words, however, we get additional improvements. Adopting the both bilingual models (BM) along with a factored LM, we improved the BLEU score further leading up to the best score of 30.57 for the test set.

5.3 English - German

In addition to the experiments on English to Romanian, we also evaluated the models on the task of translating English News to German. For the English to German system, we use three factors on the source side and four factors on the target side. In English, we used the surface forms as well as automatic word cluster based on 100 and 1,000 classes. On the target side, we used fine-graind POS-tags generated by the RFTagger (Schmid and Laws, 2008), in addition to the factors for the source side.

The experiments using only the scores of the model are summarized in Table 5. In this experiment, we analyzed a word based- and a factored language models as well as bilingual models. As described in section 4.3, the difference between the language model and the bilingual model is that the latter uses the source side information as additional factor.

Table 5: *English - German Single Score*

Model	Single
LM 5K	20.92
Factored LM 5K	21.69
BM 5K	21.33
Factored BM 5K	21.92

Using only the word-based language model we achieved a BLEU score of 20.92. Deploying a factored language model instead, we can improve the BLEU score by 0.7 BLEU points to 21.69. While we achieved a score of 21.33 BLEU points by using a proposed bilingual model, we improved the score up to 21.92 BLEU points by adopting all factors for the bilingual model.

Table 6: *English-German Language Model*

Model	Conf1	Conf2
Baseline	23.25	23.40
Factored LM 5K	23.63	23.77
Factored BM 5K	23.43	23.48

In addition to the analysis on the single model, we also evaluated the model's influence by combining the model with the baseline features. We tested the language model as well as the bilingual model on two different configurations. Adopting the factored language model on top of the baseline features improved the translation quality by around 0.4 BLEU points for both configurations, as shown in Table 6. Although the bilingual model could also improve the translation quality, it could not outperform the factored language model. The combination of the two models, LM and BM, did not lead to further improvements. In summary, the factored language model improved the BLEU score by 0.4 points.

5.4 German - English

Similar experiments were conducted on the German to English translation task. For this language pair, we built models using a vocabulary size of 5,000 words. The models cover word surface forms and two automatic word clusters, which are

based on 100 and 1,000 word classes respectively. First, we will evaluate the performance of the system using only this model in rescoring. The results are summarized in Table 7.

Table 7: *German - English Single Score*

Model	Single
LM 5K	26.11
Factored LM 5K	26.96
BM 5K	26.77
Factored BM 5K	26.81

The word based language model achieves a BLEU score 26.11. Extending the model to include factors improves the BLEU score by 0.8 BLEU points to 26.96. If we use a bilingual model, a word based model achieves a BLEU score of 26.77 and the factored one a BLEU score of 26.81. Although the factored model performed better than the word-based models, in this case the bilingual model cannot outperform the language model.

Table 8: *German - English Language Model*

Model	Single
Baseline	29.33
+ Factored BM 5K	29.51
+ Factored LM 5K	29.66

In a last series of experiments, we used the scores combined with the baseline scores. The results are shown in Table 8. In this language pair, we can improve over the baseline system by using both models. The final BLEU score is 0.3 BLEU points better than the initial system.

6 Conclusion

In this paper, we presented a new approach to integrate additional word information into a neural network language model. This model is especially promising for morphologically rich languages. Due to their large vocabulary size, additional information such as POS-tags are expected to model rare words effectively.

Representing words using factors has been successfully deployed in many phrase-based machine translation systems. Inspired by this, we represented each word in our neural network language model using factors, facilitating all available information of the word. We showed that using the factored neural network language models can improve the quality of a phrase-based machine translation system, which already uses several factored language models.

In addition, the presented framework allows an easy integration of source side information. By incorporating the alignment information to the source side, we were able to model the translation process. In this model, the source words as well as the target words can be represented by word factors.

Using these techniques, we are able to improve the translation system on three different language pairs of the WMT 2016 evaluation. We performed experiments on the English-Romanian, English-German and German-English translation task. The suggested technique yielded up to 0.7 BLEU points of improvement on all three tasks.

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. The research by Thanh-Le Ha was supported by Ministry of Science, Research and the Arts Baden-Württemberg.

References

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2*, NAACL-Short '03, pages 4–6, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 â07, pages 129–136, New York, NY, USA. Acm.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.

J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52st Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, volume 1, pages 1370–1380, Baltimore, Maryland, USA.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2014. Lexical Translation Model Using a Deep Neural Network Architecture. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT14)*, Lake Tahoe, CA, USA.

Thanh-Le Ha, Eunah Cho, Jan Niehues, Mohammed Mediani, Matthias Sperber, Alexandre Allauzen, and yAlex Waibel. 2016. The karlsruhe institute of technology systems for the news translation task in wmt 2016. In *Proceedings of the ACL 2016 First Conference on Machine Translation (WMT2016)*.

Teresa Herrman, Jan Niehues, and Alex Waibel. 2015. Source Discriminative Word Lexicon for Translation Disambiguation. In *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT15)*, Danang, Vietnam.

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.

Radu Ion, Elena Irimia, Dan Stefanescu, and Dan Tufis. 2012. Rombac: The romanian balanced annotated corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet U?ur Do?an, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL*, pages 868–876.

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 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies (NAACL)*, pages 39–48. Association for Computational Linguistics.

Nicholas Léonard, Sagar Waghmare, Yang Wang, and Jin-Hwa Kim. 2015. rnn : Recurrent library for torch. *CoRR*, abs/1511.07889.

Mohammed Mediani, Eunah Cho, Jan Niehues, Teresa Herrmann, and Alex Waibel. 2011. The KIT english-french translation systems for IWSLT 2011. In *2011 International Workshop on Spoken Language Translation, IWSLT 2011, San Francisco, CA, USA, December 8-9, 2011*, pages 73–78.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

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

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.

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.

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, Scotland, United Kingdom.

Jan Niehues, Quoc Khanh Do, Alexandre Allauzen, and Alex Waibel. 2015. Listnet-based MT Rescoring. *EMNLP 2015*, page 248.

Franz Josef Och. 1999. An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, Bergen, Norway.

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, Japa.

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.

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 2007)*, Skövde, Sweden.

Helmut Schmid and Florian Laws. 2008. Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. In *Proceedings of the 22nd International Conference*

on Computational Linguistics, Manchester, United Kingdom.

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

Martin Sundermeyer, Ilya Oparin, Jean-Luc Gauvain, Ben Freiberg, Ralf Schluter, and Hermann Ney. 2013. Comparison of feedforward and recurrent neural network language models. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8430–8434. IEEE.

Ke Tran, Arianna Bisazza, Christof Monz, et al. 2014. Word translation prediction for morphologically rich languages with bilingual neural networks. Association for Computational Linguistics.

Joern Wuebker, Stephan Peitz, Felix Rietig, and Hermann Ney. 2013. Improving statistical machine translation with word class models. In *Conference on Empirical Methods in Natural Language Processing*, pages 1377–1381, Seattle, WA, USA, October.