Adaptation and Combination of NMT Systems: The KIT Translation Systems for IWSLT 2016

Eunah Cho, Jan Niehues, Thanh-Le Ha, Matthias Sperber, Mohammed Mediani, Alex Waibel

Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

In this paper, we present the KIT systems of the IWSLT 2016 machine translation evaluation. We participated in the machine translation (MT) task as well as the spoken language language translation (SLT) track for English→German and German→English translation.

We use attentional neural machine translation (NMT) for all our submissions. We investigated different methods to adapt the system using small in-domain data as well as methods to train the system on these small corpora. In addition, we investigated methods to combine NMT systems that encode the input as well as the output differently. We combine systems using different vocabularies, reverse translation systems, multi-source translation system. In addition, we used pre-translation systems that facilitate phrase-based machine translation systems.

Results show that applying domain adaptation and ensemble technique brings a crucial improvement of 3-4 BLEU points over the baseline system. In addition, system combination using n-best lists yields further 1-2 BLEU points.

1. Introduction

The Karlsruhe Institute of Technology participated in the IWSLT 2016 Evaluation Campaign with systems for English→German and German→English. For both directions, we participated in machine translation and spoken language translation tracks. All submitted systems use the framework of attentional neural machine translation [1] extended with further features.

In this evaluation campaign, we investigated the importance of domain adaptation, where ensemble technique is deployed for this scenario. In addition to adaptation, we train further systems with different architectures and combine them by n-best rescoring. One of the systems uses pre-translation. In this system, we utilize pre-translations from a phrase based machine translation (PBMT) system in order to handle the rare word problem of NMT. The pre-translation is then used as an additional input to the NMT system. Furthermore, we used a system utilizing multi-lingual learning. The systems are, along with the ensembled systems of adaptation, combined using the n-best lists.

This paper is structured as follows. In Section 2, we describe the adaptation technique we used in order to fit the models better to the domain. A brief explanation of the pre-translation and multi-lingual learning will be given in Section 3 and Section 4 respectively. How the different systems are combined will be described in Section 5. Special preprocessing for SLT input will be described afterwards in Section 6, followed by the results of experiments and detailed analysis on the techniques used throughout in this work. Finally Section 8 concludes our discussion.

2. Adaptation

One of the main challenges of the IWSLT evaluation is to adapt the MT system towards the target domain. While relatively large out-of-domain corpora are available for training, the in-domain data is often limited. For the TED task, only around 200K sentences of in-domain data are available.

Motivated by the work of [2] and [3], we first trained the NMT system on the out-of-domain data. Once the BLEU scores converge on validated set, we used the best model trained on the out-of-domain data to resume the training on the in-domain data. An in-domain validation set is used for this training. While dropout did not have a big influence during the training on the large out-of-domain corpus, it was very important when training on the in-domain data. The detailed discussion on the results will be given in Section 7.2.

2.1. Ensemble

An ensemble of different models can often improve the performance of an NMT system. In a recent system [4], it was shown that the combination of models saving different time steps of the training $M_{T1}, \ldots, M_{Tn}$ was very successful.

In this evaluation campaign, we analyzed different ways to adapt this method for the domain adaptation scenario. In the first method, we take the best model trained on the out-of-domain corpus $M_{T}*$. The training is continued on the in-domain data and the intermediate models $M^A_{T1}, \ldots, M^A_{Tm}$ are stored. Then these models are ensembled to generate the final model. In the second strategy, on the other hand, all models $M_{T1}, \ldots, M_{Tn}$, trained on the out-of-domain data, are adapted separately on the in-domain data. The final

model is the ensemble of all the separately adapted models.

In addition, it might also be helpful to use baseline models in the ensemble. This approach can be encouraged further when the in-domain data and the test data are not expected to match precisely, as in the MSLT task. The details of the MSLT task and corpus is explained in [5].

3. Pre-translation

One of the main problems of current NMT system is its limited vocabulary [6], causing challenges when translating rare words. While the overall performance of NMT is significantly better on many tasks compared to SMT [7], the translation of words seen only a few times is often not correct. In contrast, PBMT is able to memorize a translation it has only seen once in the training data. Therefore, we tried to combine the advantages of NMT and PBMT using Pre-translation as described in [8].

In the first step, we translate the source sentence f using the PBMT system generating a translation e^{SMT}. Then we use the NMT system to find the most probable translation e^* given the source sentence f and the PBMT translation e^{SMT}. Thus, we create a mixed input for the NMT system consisting of both sentences by concatenating them. This scheme, however, may lead to errors when the source and target languages have a same word in surface, but with different meanings, i.g. *die* in English is a verb, while it is an article in German. In order to prevent such errors, we use a separate vocabulary for each language. An overview of the system can be found in Figure 1.

Using the byte-pair encoding (BPE) of the input [9], we are able to encode any input words as well as any translation of the PBMT system. Thereby, the NMT is able to learn to copy translations of the PBMT system to the target side.

For both translation directions, we used the pre-translation from a PBMT system. The detailed description on the PBMT systems for both directions can be found in [10]. The final systems without rescoring are used for generating the pre-translations.

4. Mix-source multilingual system

In [11], a multilingual NMT system shows that additional information from other languages can improve a single NMT system and produce better translations. When the encoder of an NMT system considers words across languages as different words, with a well-chosen architecture, it is expected to be able to learn a good representation of the source words in a joint embedding space in which words carrying similar meaning would have a closer distance to each others than those are semantically different. In turn, the shared information across source languages could help improve the choice of words in the target side. For example, the word *Flussufer* in German and the word *bank* in English should be projected into two points in a proximity of that joint embedding space. And that information might help to choose the French word *rive* over *banque*.

To make an attention NMT for single language pair translation be able to used as a multilingual NMT that shared the common semantic space, [11] conducted *an additional preprocessing step*, namely *language-specific coding*. Basically, some language code are appended to every word in source and target sentences to indicate the original language the word belongs to before passing to the training process of the NMT system. For example an English-German sentence pair *excuse me* and *entschuldigen Sie* being *language-specific coded* becomes *_en_excuse _en_me* and *_de_entschuldigen _de_Sie*. By doing so, they can train a *single* multilingual system that translates from several source languages to one or several target languages. For example, if we have N English-German sentence pairs and M French-German sentence pairs already *language-specific coded*, we can train a single NMT system with a parallel corpus of $N + M$ sentence pairs. Then we can use the trained model to either translate from English or from French to German.

The aforementioned multilingual NMT can be used wisely as a novel way to utilize the monolingual data, which is not a trivial task in NMT systems. Particularly, if we want to translate from English to German, we can use some monolingual German data, either the monolingual part of the parallel corpus or some part of other corpus available only in German, as an additional German-German data similar to the way we ultilize the French-German parallel corpus. Thus, the encoder is shared between the source and the target languages (English and German), and the attention is also shared across languages to help the decoder selects better German words in the target side. The systems implemented this idea is referred as a *mix-source* system and it is shown in Figure 2.

For this evaluation, we apply the idea of that multilingual NMT approach in English-German direction in order to make use of the German monolingual corpus and gain additional improvements.

5. System Combination

Combination of different neural networks often leads to better performance, as shown in various applications of neural networks and previous NMT submissions in evaluation campaigns [7]. In our previously mentioned systems in Section 2, for example, different models are ensembled during decoding. While this is a very helpful technique, it has a potential drawback that it can only be performed easily for models using the same input and output representations.

In order to further extend the variety of models, we combine the output of several of ensemble models by an n-best list combination. We first generate an n-best list from all or several of the models, where each of these models is already an ensemble of several models. In our experiments, we used $n = 50$ for the n-best list size. Then, we combine the n-best lists into a single one by creating the union of the n-best lists. Since every model only generated a subset of the joint

Figure 1: Pre-translation

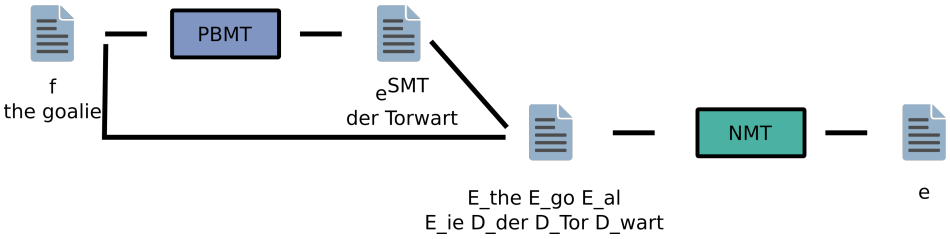
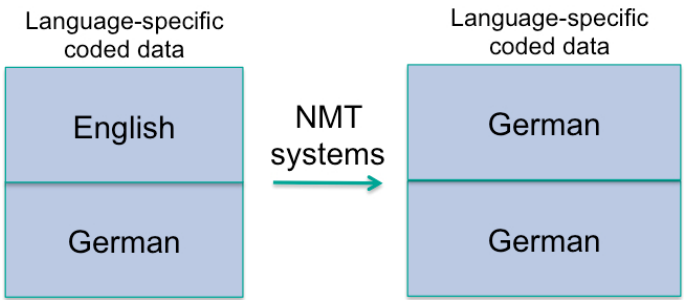


Figure 2: The English→German *mix-source* system



list, we rescored the joint list by each model. Finally, we used a combination of all the scores to select the best entry for every source sentence.

For systems to be combined, we used the baseline NMT system as well as the pre-translation and multi-lingual systems. For some of these systems, we also combined systems using different BPE sizes. In addition, we also used a system that generates the target sentence in the reversed order [12, 13, 4]. Finally, we used also the NMT systems for the reverse translation direction to rescore the n-best list. Therefore, we swapped the source and target language in the n-best list and rescored this list with the translation system of the reverse translation direction. This means that instead of n translation of one sentence, we now have n source sentence where the translation is always the same. The we used this additional probability as an additional feature.

After joining the n-best lists and rescoring the joint n-best lists using the different systems, we have k scores for every entry in the n-best lists. Each score is a length-normalized log-probability.

6. Preprocessing for Speech Translation

Many state-of-the-art automatic speech recognition systems do not generate punctuation marks or reliable case information. Using the raw output of such automatic speech recognition (ASR) systems as an input to an MT system causes performance drop. In this evaluation campaign, we used monolingual translation systems for each source language to augment proper punctuation marks and sentence boundaries [14]. The monolingual translation system *translates* non-punctuated test data into a punctuated one. During this process, case information is corrected as well.

The parallel data for training consists of lower-cased source side without any punctuation and true-cased target side with all punctuation marks. Note that the source side language and target side one are the same, except for the punctuation and case information. The training data is randomly segmented, so that the location of segment boundaries and different punctuation marks is well-distributed throughout the corpus.

The monolingual translation system was applied to all official SLT track directions. For MSLT track of English→German and German→English SLT, segment boundaries are given. Therefore, the monolingual translation system is used to predict punctuation marks within the boundaries. For TED track of English→German SLT, however, no segment boundaries are given. Therefore, we applied the monolingual translation system to resegment sentence boundaries as well. For this, we used a sliding window of length 10 to observe each word in various contexts as described in [14].

Both English and German systems are trained on EPPS, TED, NC and noise-filtered common crawled data. Each language corpus sums up to 3.9 million sentences. Models used in the phrase-based monolingual translation systems for English and German are similar. We used GIZA++ [15] to obtain the alignment between non-punctuated, lower-cased text and punctuated, cased text.

The 4-gram word-based language model is built on the entire punctuated data using the SRILM Toolkit [16]. A bilingual language model [17] is used, along with a a 9-gram

part-of-speech-based language model. TreeTagger [18] was used to obtain POSs for both languages. In addition, we train a $1,000$-class cluster on the punctuated data. A 9-gram language model is built on the cluster codes. The models were optimized on the official test set of IWSLT evaluation campaign in 2013.

7. Results and Analysis

In this section, we present a summary of our experiments we have carried out for the IWSLT 2016 evaluation. All the reported scores are case-sensitive BLEU scores.

7.1. Baseline Systems

All our NMT systems are built using the framework `nematus`[1]. We used sub-word units using BPE as described in [9]. For both languages, we apply the BPE operations at 40K (represented as *SmallVoc* throughout this paper) and 80K (*BigVoc*) on the joint source and target data depending on the configurations, which are then combined.

Long sentences whose sentence length exceeds 50 words are exempted from the training. We use minibatch size 80 and sentences are shuffled within every minibatch. Word embedding of size 500 is applied, with hidden layers of size 1024. Dropout is applied at every layer with the probability 0.2 in the embedding and hidden layers and 0.1 in the input and output layers. Our models are trained with Adadelta [19] and the gradient norm is clipped to 1.0. We use a beam search for decoding, with the beam size of 12.

The baseline systems were trained on the WMT parallel data. For both languages, this consists of the EPPS, NC, CommonCrawl corpus. In addition, we randomly subsampled a same size corpus from the monolingual news crawled corpus and created an additional pseudo parallel corpus as described in [12]. As in-domain data, we used the TED corpus.

Throughout this paper, validation data denotes the newstest13 set, while test data the newstest14 set. For the single models, we apply the early stopping based on the validation score.

7.2. Results of Adaptation

Our first line of experiment is dedicated on establishing the effect of training on a large corpus (e.g. out-of-domain data) or a small corpus (e.g. in-domain data). For English to German, for example, we trained one system only on the out-of-domain data and another only on the in-domain data independently. The results are shown in Table 2. On the other hand, we experimented on the impact of domain adaptation on the German to English systems. Namely, we compared the system trained only on the out-of-domain data against the adapted model.

One important observation is the usefulness of dropout. We notice that using dropout in large and out-of-domain data does not help while an enormous improvement is observed when we use dropout in much smaller and in-domain data. Also when we look at the situation where we when continuing the training on the in-domain data, dropout is very important. In this case, we cannot improve the model, if we do not use dropout. The system overfits to the training data and the performance on the unseen test data even drops. In contrast, if we use dropout in the adaptation phase, we can improve the translation quality by 3 BLEU points.

One reason could explain this is that dropout helps to reduce overfitting when training on the small data. On the large and well-covered data, however, it introduces unnecessary noises and does not bring any positive impact.

Table 1: Effect of using dropout on German→English

System	System	Valid	Test
Baseline	Dropout	30.96	26.77
	No Dropout	32.53	27.43
Adapted	Dropout	35.35	30.66
	No Dropout	30.56	25.18

Table 2: Effect of using dropout on English→German

System	System	Valid	Test
Baseline	Dropout	24.87	21.03
	No Dropout	25.44	22.24
In-domain	Dropout	24.35	20.62
	No Dropout	19.86	17.75

As shown in Table 1, the adaptation to the TED domain is very helpful for the German-English translation system. Table 3 confirms the essential of adaptation in our English→German configurations. The non-adapted configurations are trained on the out-of-domain concatenation corpus without dropout, and the adapted ones are continuously trained on the in-domain TED data with dropout in every layer of the networks.

Another interesting finding from Table 3 is while the configuration trained on the large corpus is not beneficial by using bigger vocabularies, its adaptation on the small, in-domain data brings a great improvement over the adapted configuration using small vocabularies (e.g. 26.72 versus 24.13 on *tst2014* in term of BLEU scores).

In another line of research, we analyze the influence of the baseline model on the adapted final model. We measure the performance of the baseline and adapted systems, when different iterations of training are applied for the baseline training. Therefore, the experiments should answer the question if it is helpful to train the baseline model for many iterations or if an initial model is sufficient for initializing the

[1] `https://github.com/rsennrich/nematus`

Table 3: Effect of adaptation on English → German NMT configurations

Configuration	No adaptation			Adaptation		
	Valid	Test	MSLT	Valid	Test	MSLT
SmallVoc	25.74	22.54	35.06	28.08	24.13	36.27
BigVoc	25.95	22.51	35.93	31.08	26.72	37.61

adaptation process. The results are summarized in Table 4.

Table 4: Training length of baseline model

Iteration	Baseline		Adapted	
	Valid	Test	Valid	Test
300K	31.54	27.15	34.85	29.97
450K	31.97	27.56	35.16	29.95
600K	32.67	28.35	35.35	30.66

We trained the baseline model for 300K, 450K and 600K iterations. As shown in the table, this leads to an improvement of 1.2 BLEU points on the initial model trained only on the out-of-domain data. If we adapt these models by continuing training on the in-domain data, we can improve by 2 to 3 BLEU points. While the difference between the different models is lower, the model trained for 600K iterations is still 0.6 points better. In order to achieve the best performance, thus, it is important to train the baseline model till convergence.

After analyzing the design decision when training an adapted model, we performed further experiments for the ensemble of different models. [3] shows that an ensemble of various adapted configurations is usually helpful. Ensembling also helps in our cases as showed in Table 5 for German→English and Table 6 for English→German.

Table 5: Ensemble of German→English adapted models

System	Valid	Test TED	Test MSLT
Baseline	32.67	28.35	33.21
+ Adapted	35.35	30.66	33.40
+ Ensemble (3 Models)	35.76	31.00	34.25
+ 1 Baseline	36.72	31.99	35.82
+ 2 Baseline	36.84	31.97	37.11
+ 3 Baseline	36.93	31.69	37.50
+ 4 Baseline	36.75	31.49	37.77

In German→English, we ensemble three adapted models. This could improve the translation quality by only 0.3 BLEU points. By further adding up to three baseline models, we get further improvements by 1 BLEU point on the validation set and 0.7 on the test set. As shown in the final results in Table 8 and 9, this finding was not consistent throughout all models. However, the combination of adapted and non-adapted models is very useful for MSLT data, which does not exactly match our in-domain (TED) data.

Table 6: Ensemble of adapted English→German models

System	Valid	Test TED	Test MSLT
Baseline	27.74	24.39	35.19
+ Adapted	29.89	25.46	36.32
+ Ensemble A4B0	30.58	26.09	37.28
+ Ensemble A3B1	30.83	26.40	38.62
+ Ensemble A2B2	30.99	26.10	39.11
+ Ensemble A1B3	30.10	26.00	38.88

In English→German, we conduct the similar experiments of ensembling using the mix-source system. All of the ensembles include 4 models, and they are different on which adapted models and which baseline models are chosen. For example, *Ensemble A4B0* means the best four adapted models and none of the baseline models are chosen to be ensembled. Likewise, *Ensemble A2B2* means the best two adapted models and the best two baseline models are chosen to be ensembled. Similar in the German→English case, we observe that although the baseline configurations performed much worse than the respective adapted ones, the ensemble of some baseline and adapted models sometimes works better than the ensemble of all adapted models (*Ensemble A2B2* and *Ensemble A3B1* are better than *Ensemble A4B0* both on TED and MSLT tasks). The improvements when using ensembles are considerable in this case: almost 1 BLEU points on TED task and 2.79 BLEU points on MSLT task.

Table 7: Ensemble of adapted Models

Iteration	Only Adapt		Adapted + Baseline	
	Valid	Test	Valid	Test
1 Adaptation	35.76	31.00	36.93	31.69
MultiAdapt 30K	35.67	31.14	37.01	31.70
MultiAdapt 150K	36.04	30.97	36.89	31.81

In the last experiment, we trained one baseline model and adapted it by continue training on the in-domain data.

During the adaptation, we stored different models which we combined in the ensemble model. The results for German→English are shown in the first row in Table 7. A different strategy would be to take different baseline models and apply the adaptation on each of them. The results are shown in the next two rows. As seen in the results, this does not really improve the translation quality. Namely, it seems to be sufficient to adapt one baseline model.

7.3. German→English

As described in Section 5, we combined different, already ensembled systems by rescoring. The initial systems for the TED task are shown in the first several rows of Table 8. This table also shows how many systems are ensembled for each combined system. As the initial systems, we used a baseline system (*SmallVoc*), a system that generated the target sentence in the reverse order (*SmallVoc.rev*), a pre-translation system [8] and a system using a 80K vocabulary (*BigVoc*). The best performance is reached by the BigVoc translation system.

Table 8: System combination TED

System	Base	Adapt	Valid	Test
(1) SmallVoc	3	3	37.01	31.74
(2) SmallVoc.rev	1	3	36.56	31.28
(3) Pre-translation	0	4	36.43	31.41
(4) BigVoc	0	4	37.50	32.41
Sum (1+2+3+4)	4	14	38.95	33.40
(5) Inverse	0	4	34.43	29.11
ListNet (1+2+3+4+5)	4	18	39.22	33.69

Then we generated the joint n-best lists and rescored the joint system using each system, represented as *Sum* in the table. A log-linear combination of all systems with equal weights for each system can improve the performance by 1 BLEU point to 33.40.

Table 9: System combination MSLT

System	Base	Adapt	Test
(1) SmallVoc	4	3	37.90
(2) SmallVoc.rev	4	3	38.72
(3) Pre-translation	4	0	38.33
(4) BigVoc	4	2	38.80
Sum (2+3+4)	8	14	40.75
(5) Inverse	0	4	34.27
Sum (2+3+4+5)	16	12	40.93

In a second system, we also rescored the n-best list with a translation system from English to German, named *Inverse* in the table. This system performed significantly worse than all other systems and reaches a BLEU score of 29.11. A linear combination using equal weights on all systems did not improve the performance. If we, in contrast, train the weights using the ListNet algorithm [20], we are able get further improvements of 0.3 BLEU points.

For the MSLT test set, we performed similar experiments. In this task, we face the problem that we do not have a development set. Since we saw in the performance on the development and test data correlate quite well, we selected our final submission based on the performance on the dev test set. As shown in Table 9, for these systems it was beneficial to use more baseline systems for the ensemble of each combination. Again, we could improve the performance by 2 BLEU points by using a combination of three system combinations. The *Inverse* translation system performed worse, similar to the experiments on TED. Due to lack of additional development data for this task, we could not train the weights using the ListNet-based rescoring. When using a linear combination with equal weights, we are able to improve the performance by additional 0.2 BLEU points.

7.4. English→German

In the TED task, for each *SmallVoc* and *BigVoc* configurations, we also train and adapt the corresponding reversed (*.rev*) and mix-source (*.mixs*) systems with the aforementioned adaptation scheme. The pre-translation systems (*Pre-translation* and *Pre-translation.mono*) from the *SmallVoc* are also trained and adapted. *Pre-translation.mono* indicates an additional monolingual data is used for training the system. For each system, we conduct several ensembles as described in Section 7.2 and choose the best ensemble based on the performance evaluated on the valid set. Table 10 reports the scores of those best ensembled systems.

Table 10: English→German TED translation

System	Base	Adapt	Test
(1) SmallVoc	2	2	26.63
(2) SmallVoc.rev	2	2	26.28
(3) SmallVoc.mixs	1	3	26.40
(4) Pre-translation	0	4	26.44
(5) Pre-translation.mono	0	2	27.03
(6) BigVoc	0	4	27.19
(7) BigVoc.rev	1	3	26.60
(8) BigVoc.mixs	1	3	26.31
Sum(2+4+5+6+8)	3	15	28.02

Then we generated the joint n-best lists and rescored the joint system using each system. The best system is the best log-linear combination of some individual systems with equal weights. In this TED task, the combination of 5 different systems brings an 0.83-BLEU-point improvement over the best ensembled individual system and 2.56-BLEU-improvement over the best adapted one.

We conduct similar experiments for the MSLT task. As shown in Table 11, the best ensemble is the ensemble of 2 adapted models and 2 baseline models from *SmallVoc* system, scoring 39.30 BLEU points. Again, an improvement of 1.22 BLEU points can be obtained by using a combination of four systems.

Those two best combination are our submitted systems to the evaluation campaign.

Table 11: English→German MSLT translation

System	Base	Adapt	MSLT test
(1) SmallVoc	2	2	39.30
(2) SmallVoc.rev	2	2	37.54
(3) SmallVoc.mixs	2	2	39.11
(4) Pre-translation	1	2	37.45
(5) Pre-translation.mono	1	4	38.28
(6) BigVoc	1	3	38.72
Sum(1+3+5+6)	6	11	40.52

8. Conclusions

In this paper, we described several innovative techniques that we applied to our neural machine translation systems we submitted to the IWSLT 2016 Evaluation Campaign. In this evaluation campaign, we participated in official MT and SLT tasks for English→German and German→English.

For both of the translation directions, we obtained improvements in translation performance by applying the adaptation technique. Different systems, such as the one uses pre-translation as an additional input source and the one trained with reversed target side, are combined based on n-best lists. The experiments show that reranking improves the translation performance further.

9. Acknowledgements

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.

10. References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015.

[2] T. Lavergne, A. Allauzen, H.-S. Le, and F. Yvon, "Limsi's experiments in domain adaptation for iwslt11," in *Proceedings of the 8th International Workshop on Spoken Language Translation*, 2011.

[3] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the International Workshop on Spoken Language Translation*, 2015.

[4] M. Huck, A. Fraser, and B. Haddow, "The edinburgh/lmu hierarchical machine translation system for wmt 2016," in *Proc. of the ACL 2016 First Conf. on Machine Translation (WMT16), Berlin, Germany, August*, 2016.

[5] W. D. L. Christian Federmann, "Microsoft speech language translation (mslt) corpus: The iwslt 2016 release for english, french and german," in *IWSLT*, Seattle, WA, USA, 2016.

[6] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," 2014.

[7] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation (wmt16)," in *Proceedings of the First Conference on Machine Translation (WMT)*, vol. 2, 2016, pp. 131–198.

[8] J. Niehues, E. Cho, T.-L. Ha, and A. Waibel, "Pre-translation for neural machine translation," in *the 26th International Conference on Computational Linguistics (Coling 2016)*, 2016.

[9] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2015.

[10] T.-L. Ha, E. Cho, J. Niehues, M. Mediani, M. Sperber, A. Allauzen, and A. Waibel, "The karlsruhe institute of technology systems for the news translation task in wmt 2016," in *Proc. of the ACL 2016 First Conf. on Machine Translation (WMT16), Berlin, Germany, August*, 2016.

[11] T.-L. Ha, J. Niehues, and A. Waibel, "Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder," in *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016) - To be appeared*, Seattle, WA, USA, 2016.

[12] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," 2015.

[13] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Agreement on target-bidirectional neural machine translation," in *Proceedings of NAACL-HLT*, 2016, pp. 411–416.

[14] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *Proceedings of the 9th International Workshop on Spoken Language Translation*, Hong Kong, 2012.

[15] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, 2003.

[16] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, USA, 2002.

[17] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, United Kingdom, 2011.

[18] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

[19] M. D. Zeiler, "Adadelta: an adaptive learning rate method," in *CoRR*, 2012.

[20] J. Niehues, Q. K. Do, A. Allauzen, and A. Waibel, "Listnet-based MT Rescoring," *EMNLP 2015*, p. 248, 2015.