Ensemble Learning for Multi-Source Neural Machine Translation

Ekaterina Garmash and **Christof Monz**
Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{e.garmash,c.monz}@uva.nl

Abstract

In this paper we describe and evaluate methods to perform ensemble prediction in neural machine translation (NMT). We compare two methods of ensemble set induction: sampling parameter initializations for an NMT system, which is a relatively established method in NMT (Sutskever et al., 2014), and NMT systems translating from different source languages into the same target language, i.e., multi-source ensembles, a method recently introduced by Firat et al. (2016). We are motivated by the observation that for different language pairs systems make different types of mistakes. We propose several methods with different degrees of parameterization to combine individual predictions of NMT systems so that they mutually compensate for each other's mistakes and improve overall performance. We find that the biggest improvements can be obtained from a context-dependent weighting scheme for multi-source ensembles. This result offers stronger support for the linguistic motivation of using multi-source ensembles than previous approaches. Evaluation is carried out for German and French into English translation. The best multi-source ensemble method achieves an improvement of up to 2.2 BLEU points over the strongest single-source ensemble baseline, and a 2 BLEU improvement over a multi-source ensemble baseline.

1 Introduction

It has been shown for various machine learning applications that combining multiple systems can substantially improve performance (Rokach, 2010). System combination has also been successfully applied to statistical machine translation system (SMT) (Och and Ney, 2001; Matusov et al., 2006; Schwartz, 2008; Schroeder et al., 2009). However, system combination methods in the phrase-based (PBSMT) (Koehn et al., 2003) and hierarchical (HSMT) frameworks (Chiang, 2007) tend to be rather complex, requiring potentially non-trivial mappings between the partial hypotheses across the search spaces of the individual systems. For this reason SMT system combination is often limited to combining hypotheses from the n-best list. Alternatively, SMT systems can also be combined by processing different inputs as is the case for multilingual system combination. Unfortunately, input sentences in different languages may have very different structure, requiring elaborate methods to align sentences, which means that multilingual system combination is in practice restricted to languages with similar structures.

On the other hand, the recently emerged neural machine translation (NMT) framework offers a straightforward way to combine multiple systems. Most of the current NMT architectures (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) formalize the target sentence generation as a word sequence prediction task. At each step during sequence prediction, a translation system outputs a *full* probability distribution over the target vocabulary. Therefore, the task of NMT system combination can be cast as an ensemble prediction task and a variety of existing general prediction combination methods can be applied. While this paper focuses on word-based models, the ensemble methods discussed in this paper can be applied to character-based sequential NMT models (Ling et al., 2015) in a very similar fashion.

Ensemble prediction is frequently used in NMT. A commonly reported method is uniform weighting of the output layers, i.e., distributions over the target vocabulary, produced by different trained instances of the same NMT architecture for the same language pair (Bojar et al., 2014). We use this method as

a baseline where the variations of the same system are produced by different parameter initializations. Alternatively, it is also possible to take parameter snapshots from different training epochs (Sennrich and Haddow, 2016). Recently, a new type of ensemble has been introduced in NMT: a *multi-source ensemble* (Firat et al., 2016), which is a set of NMT systems that translate from different source languages into the same target language. The general idea behind ensembles is that different predictors are likely to produce slightly different errors for different input instances, and if their predictions are combined, the overall error is reduced. Therefore, it is essential to introduce a minimum of diversity into an ensemble. Different random initializations force the same training algorithm to converge to different local optima. Different source sentences may differ quite significantly in their structure and thus present a different training task to an NMT learning algorithm. Multi-source ensembles offer a *linguistic* source of variation for translation systems, which may range from the way particular words are translated to the way the whole sentence is structured. This is in line with the common observation that translation systems trained on language pairs with different source languages differ in their performance (Bojar et al., 2014).

Besides the linguistic interest, multi-source translation can be applied in practical real-life scenarios. Examples include multi-lingual websites, where some content has already been made available in a couple of languages (by human translators) but needs to be further translated into other languages. Another example are parliamentary proceedings, typically available in many languages. Furthermore, Firat et al. (2016) study neural multi-source translation in the context of zero-resource translation, and experimentally show that multi-source pivot-based translation improves translation quality compared to simple pivot-based translation.

In this paper, we compare ensemble combination methods and evaluate how much performance gain they provide in the multi-source translation setting. All previous approaches employing NMT ensembles do so by applying a simple linear, uniform weighting of the output probability distributions. However, it seems intuitive to assign different weights to different systems, especially for the case of multi-source translation. Firat et al. (2016) evaluate multi-source ensemble method on French-Spanish into English. All three languages, and especially French and Spanish are very similar, so for this scenario their contribution may be close to equal. In order to explore the diversity offered by linguistic variation, we chose to evaluate on English, German, and French. German is structurally substantially different from both English and French, while English and French are quite similar and are typically easily mutually translatable (Bojar et al., 2014). Here, we translate from French into English and from German into English and we expect the former translation direction to perform substantially better than the latter. As we mentioned above, the performance of translation systems can vary with respect to different aspects of translation quality, such as correct reordering or lexical translation choice. In order to combine the strengths of individual systems, we train different combination functions on a held-out data set. We consider two types of combination: global (fixed weight for every prediction instance) and context-dependent, where weights are estimated for every prediction step. The latter is more fine-grained and is in principle able to capture more linguistic nuance, but may be difficult to train due to data sparsity.

This paper proceeds as follows: In Section 2 we describe the NMT architecture and optimization parameters that we use to train our translation systems. We further describe the training data and report individual translation performances of the trained systems. In Section 3 we discuss in detail the ensemble methods that we wish to employ. We present a grid search experiment to explore the performance potential of the two types of ensemble sets (single-source with different random initializations and multi-source). The experiment confirms our intuition that linguistic diversity can be beneficial for building NMT ensembles. In Section 4 we describe two models to train an ensemble combination function: a global weighting function and a context-dependent gating network. Section 5 contains results of translation experiments with different types of ensembles and their discussion. Section 6 provides some conclusions and outlook on future work.

2 Attention-based sequence-to-sequence NMT

We use an encoder-decoder neural translation model as described in Luong et al. (2015) to train individual predictors in an ensemble. The source encoder is a four-layer unidirectional LSTM. The final hidden

	Set	N. of lines	N. of word tokens	N. of word types
(a)	train DE-EN	123,955	DE: 2,3M; EN: 2,5M	DE: 89K; EN: 41K
	train FR-EN	127,755	FR: 2,9M; EN: 2,6M	FR: 58K; EN: 41,9K
	valid DE-EN	2,052	DE: 40.3K; EN: 41.5K	DE: 6.3K; EN: 4.7K
	valid FR-EN	887	FR: 21.5K; EN: 20K	FR: 3.7K; EN: 3.1K
(b)	combin.train DE-EN-FR	19,000	DE: 372K; FR: 447K; EN: 396K	DE: 29K; FR: 22.8K; EN: 17K
	combin.valid DE-EN-FR	1,000	DE: 19K; FR: 23K; EN: 20.5K	DE: 4.3K; FR: 4K; EN: 3.5K
(c)	test DE-EN-FR	3,000	DE: 62.9K; FR: 78K; EN: 69.5K	DE: 9.3K; FR: 8.3K; EN: 6.5K

Table 1: Data statistics for (a) NMT training, (b) ensemble combination function training, and (c) testing.

states of the encoder are used to initialize the decoder, which is also a four-layer unidirectional LSTM. On top of that, at each time step i the target hidden state from the top layer h_i^t and the set of all source hidden states $\{h_1^s, ..., h_n^s\}$ are used to compute a context vector c_i, where n is the length of the source sentence. We use the global attention mechanism with the *dot score* function from Luong et al. (2015):

$$c_i = \sum_{j=1}^{n} \text{softmax}(\textit{score}(h_i^t, h_j^s))h_j^s \tag{1}$$

$$\textit{score}(h^t, h^s) = (h^t)^\top h^s \tag{2}$$

Finally, the last (non-recurrent) hidden state $\widetilde{h_i}$ is computed to produce the output layer y_i, which in turn is used to compute a probability distribution over the target vocabulary by applying the softmax function:

$$\widetilde{h_i} = \tanh(W_c[c_i; h_i^t]) \tag{3}$$

$$y_i = \text{softmax}(W_y\widetilde{h_i}) \tag{4}$$

In the following two subsections we describe how we train the translation systems, which are later used as part of an ensemble during beam decoding, and how they perform individually.

2.1 Training details

2.1.1 Data

We consider a multi-source scenario based on French, German, and English. We chose these languages to introduce diversity into ensembles: German is structurally substantially different from both English and French, while English and French are structurally similar and are typically easily mutually translatable (Bojar et al., 2014). We expect a French-English system to perform better than German-English. But also, given the linguistic intuition about the structural differences between these translation pairs, we hope that the two systems compensate for each other's weaknesses when used in an ensemble.

To ensure that the only distinguishing factor between different language pairs is the source language, we chose training data which is to a large extent parallel across all three languages, i.e., it is a trilingual parallel text with small bilingual parts. To this end, we train all of our systems on the TED talks data set (Cettolo et al., 2012). All available data is split into a training and a validation set to train individual NMT systems, a training and a validation set to train a combination function for ensembles, and a test set for the final evaluation. The training data for learning the ensemble combination function and the test set should necessarily be fully parallel (tri-parallel). Therefore, we extracted our test set from the available trilingual data and did not use the test sets provided by (Cettolo et al., 2012) since they are not parallel across all three languages. Of course, the test set does not overlap with the training data. Table 1 provides some statistics of the training data.

2.1.2 Network and optimization details

We set the size of all embeddings and hidden layers to $1,000$. We use LSTM units for the recurrent hidden states and apply dropout with a probability of 0.2 (Luong et al., 2015). We make sure that the output (target vocabulary) layer is exactly the same for all NMT systems in an ensemble. To this end we precomputed the intersection of the target vocabularies for the respective language pairs in an ensemble.

system	BLEU	MET EOR
de→ en best	20.58	49.16
de→ en mean	20.31 ± 0.34	48.88 ± 0.33
fr→ en best	27.80	56.91
fr→ en mean	27.03 ± 0.87	56.05 ± 0.82

Table 2: Translation results for individual NMT systems. Decoding beam size is equal to 20. For each language pair we trained four NMT systems with different weight parameter initializations. For each pair we provide the best score and the mean score with standard deviation.

We rank the words in the intersection by their summed frequencies in order to select the top n words for the output later and map the remaining words to $\langle unk \rangle$. For the source sides, we applied the same procedure except for computing the intersection. For French and German we set the vocabulary size to $35,000$ and for English to $24,000$.

For network training we use a Neural MT system Tardis implemented in Torch.[1] All parameters are uniformly initialized, except for the embeddings which were initialized by sampling from a Gaussian with unit variance. Each translation system is trained for 20 epochs. We use SGD with mini-batches of size 20 with a learning rate of 1 and a decay rate of 0.8 after the fifth epoch. During training we limit the lengths of predicted sequences to 50 tokens.

2.2 Translation experiments with individual systems

For each language pair we train four systems by sampling different initial parameter values. Table 2 summarizes the performances of the individual systems. In all of the translation experiments the beam decoding size was set to 20. We evaluate performance with BLEU (Papineni et al., 2002) and METEOR (Lavie and Denkowski, 2009). The first thing to notice is that distributions for both metrics are consistent.

Since we focus on ensemble translation, we are interested in how diverse the translation systems are in their performance. First, we can see that the two language pairs have different degrees of internal variation. The performance variance of German-English is smaller than that of French-English. Second, we see differences between the two source languages when translating into the same target language. These differences are much higher than those within one language pair. This raises the question how helpful a source language with a significantly lower performance will be in a multi-source ensemble.

3 Ensemble prediction in NMT

Generally speaking, in order to specify an ensemble method, one needs to specify how a set of predictors is induced and how they are subsequently combined to make joint predictions (Rokach, 2010). For the construction of the set of predictors, it is essential that they make diverse predictions, to decrease the prediction error. Our goal in this paper is two-fold: First, understand how different ensemble induction methods influence the quality of translation. Second, find the optimal way to combine individual predictors into an ensemble. In Section 2.2 we discussed diversity across different NMT systems. In the remainder of this section we describe the induction and combination methods we evaluate in this paper. We also present experimental analysis of achievable translation improvements.

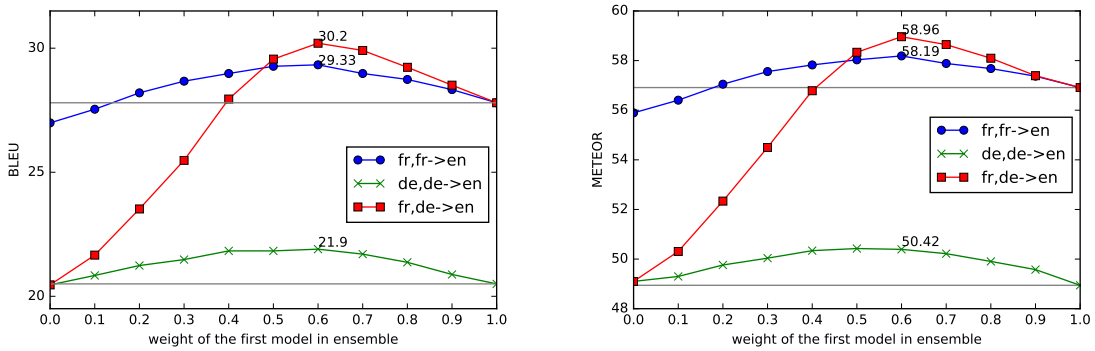
In this paper we consider two ways to induce an ensemble of translators:

1) different random initializations of NMT parameter values;

2) using semantically equivalent source sentences in different languages to translate into the same target language (translation systems with different source languages but the same target language).

The second method can be seen as different hidden state initializations of a (trained) NMT decoder. Different languages encode the same information in structurally different ways, which may influence the way the decoder is able to infer the translation from that.

[1] https://github.com/ketranm/tardis



(a) BLEU scores for Fr, De into En ensembles.

(b) METEOR scores for Fr, De into En ensembles.

Figure 1: Results of a 2-ensemble parameter sweep for the two types of ensemble induction. The x-axis represents the value of the first combination weight w_1. Number-marked points are the maximal observed scores for a given ensemble. The horizontal gray lines represent the scores of individual NMT systems used within the ensembles.

	stronger system in ensemble		uniform		best combination	
ensemble	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
fr_1,fr_2→en	**27.8**	**55.8**	29.2	58.0	29.3	58.1
de_1,de_2→en	20.5	49.10	21.8	50.4	21.9	50.4
fr,de→en	**27.8**	**55.8**	**29.5**	**58.3**	**30.2**	**58.9**

Table 3: Summary of the grid search of the scalar combination weights

In NMT the decision of which word to predict is based on the output layer and therefore we have to combine the output layers (Equation 4) of individual translators to obtain an ensemble prediction (Equation 5). The word thus predicted by an ensemble is then fed as input at the next prediction step in a sequence-to-sequence model.

$$y^{\mathcal{E}} = comb(y^1, ..., y^m) \tag{5}$$

We would like the method to be applicable to a situation where a trilingual parallel corpus, i.e., the corpus needed to train a multi-source combination function, is a scarce resource, which is a realistic assumption. Therefore we are interested in combination functions with a small number of trainable parameters. In our approach, we concentrate on *scalar* prediction combination: $comb(y^1, ...; , y^m) = w_1 y^1 + ... + w_m y^m$, where $\sum_i w_i = 1$ are scalar weights. In addition to the methods described in this paper we also investigated geometric mean combination ($\sqrt[m]{w_1 y^1 \cdot ... \cdot w_m y^m}$), but the resulting ensemble system under-performed the stronger individual system of the ensemble, therefore in the rest of the experiments we proceeded with the arithmetic mean function only.

Both single-source ensembles with different initializations (Sutskever et al., 2014) and multi-source ensembles have been used before (Firat et al., 2016). However all of the previous approaches use simple, uniform weighting. We refer to this method as *uniform combination*, as it does not assume anything about the contributions of the individual predictors. We perform grid search over the global combination weights[2] $\langle w_1, w_2 \rangle$ for a two-element ensemble (for both ensemble induction methods) over our test set with a step size of 0.1; see Section 2.1.1 for data and system setup.

The results of the grid search experiments are presented in Figure 1 and summarized in Table 3. We observe an increase in performance for both metrics for all ensembles. Moreover, we see that the metric scores are higher in the region of 0.5, which justifies uniform ensemble combination. At the same time, none of the graphs are completely symmetric: the highest scores are achieved with a weight value of 0.6 or 0.7 assigned to the stronger system in an ensemble. This result is intuitive, and it suggests

[2]I.e., a weight value is fixed for every instance in the test set.

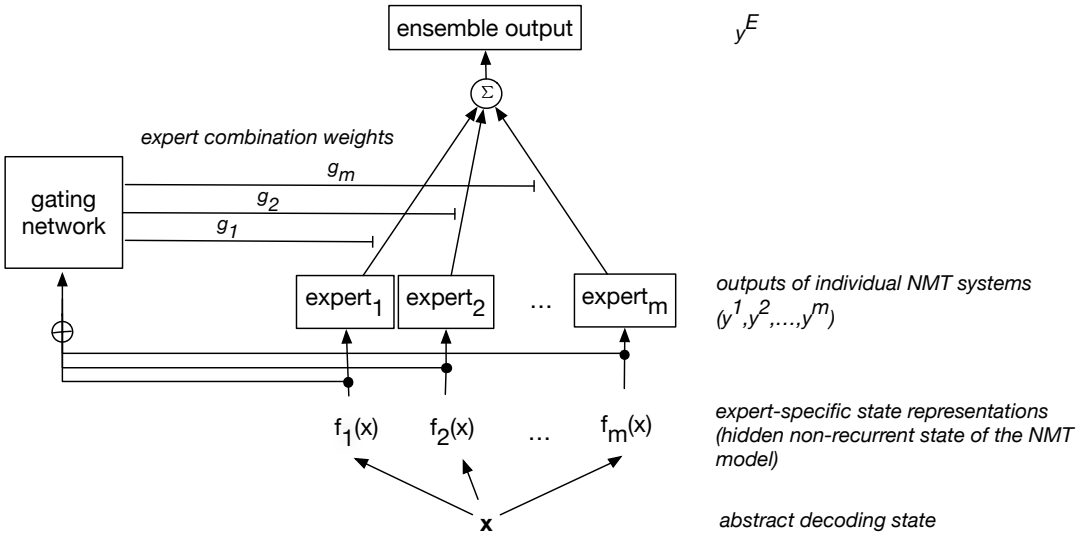


Figure 2: Mixture of NMT experts used to make context dependent translation prediction.

to investigate a combination method that could distinguish between the relative contributions of the individual members of an ensemble. We will distinguish between two kinds of combination functions: global and context-dependent. The former combines NMT predictions in the same way for every input at every decoding time step. The latter can combine predictions differently depending on the current context during decoding. We describe the corresponding learning methods in Section 4.

The second major finding of our parameter sweep is that the multi-source ensemble gives a higher upper bound performance than single-source ensembles, even though one of the ensemble members is substantially weaker in its individual performance. This finding reinforces the original linguistic motivation for multi-source ensembles with which we can obtain improvements of up to 0.87 BLEU and 0.77 METEOR over the highest single-source ensemble result.

In the following sections we describe how we make use of the uncovered potential of the two types of ensembles. We are especially interested in making full use of the complementary strengths of systems with different source languages.

4 Combination function learning

Having established that contributions of individual systems towards a correct prediction in single-source and multi-source ensembles are not equal, we develop an approach that is capable of training a function that can combine them optimally. For the case of multi-source ensembles the combination training set is a trilingual set consisting of $19,000$ lines (see Section 2.1.1 for details). We deliberately chose a small data set to establish how applicable the method is in a low-resource scenario. We use the same training set to train single-source ensembles. In this section we present two kinds of combination models, as well as their training details.

First, when a scalar combination vector is fixed for every prediction step, we refer to it as *global combination*. The optimized function is a vector $\langle w_1, ..., w_m \rangle$, where m is the size of the ensemble set. We train it with AdaGrad (Duchi et al., 2011) for 10 epochs with a learning rate of 0.001.

Second, we explore a more fine-grained combination method, where the contributions of individual predictors are assessed based on the decoding state. We adapt a mixture of experts model (Jacobs et al., 1991) to learn the *context-dependent combination*. The original mixture of experts model works as follows: We have a set of experts (predictors) and an input x, which is fed to each of the predictors. x is also fed to a gating network which outputs weights for each of the experts. The resulting prediction is a weighted sum of experts: $\mu = \sum_i g_i \mu_i$, where μ_i is the output of the i-th expert and g_i is its gating weight. Here, we realize context dependence by making use of a parameterized gating network.

Adapting the mixture of experts model (Jacobs et al., 1991) to the NMT scenario presents itself with a few challenges. NMT models are sequential and therefore the output at time step i depends on the current input word and the previous hidden state, which encodes the translation history for a *given* expert. This leads to two problems: the input representation is specific to an expert and it is also quite complex as it is a combination of hidden state and previously predicted word. We address the first problem by simply concatenating vectors which are inputs to each of the translators at time step i. There are a number of ways to address the second problem. But essentially, we would like to think of input x as some abstract decoding state corresponding to the context of the ensemble translation process at time step i.

In the first set of experiments, we opt for using the already available representations for the decoding state x, rather than formulating an explicitly, linguistically-motivated definition. Given the complex modular structure of an NMT model, there are a number of hidden states, such as the hidden recurrent states, the context vector, or the non-recurrent hidden state $\widetilde{h}$, which can be chosen to represent the decoder state which is the input to the gating network. In our approach, we use the last hidden state $\widetilde{h}$. We choose $\widetilde{h}$ because it already captures a large amount of information such as the previously predicted word, previous hidden state, and attention distribution over the source words.[3] In addition, the output layer is more directly connected to $\widetilde{h}$ than any of the states from lower layers. This is an important consideration given that the amount of training data is severely limited.

The architecture of our context-dependent combination function is presented graphically in Figure 2. The ensemble output $y^{\mathcal{E}}$ is computed as in Equation 6, where g_j is the gating weight, x represents the abstract decoding state at step i and $f^j(x)$ is its expert-specific representation (for expert j), namely $\widetilde{h}^j$:

$$y^{\mathcal{E}}(x) = \sum_j g_j \mu_j(x) \tag{6}$$

$$\begin{aligned} \mu_j(x) &= \text{softmax}(W_y f^j(x)) && (7)\\ &= \text{softmax}(W_y \widetilde{h}^j) && (8)\\ &= y^j && (9) \end{aligned}$$

g_j is the j-th output unit of the gating network computed as in Equation 10. The gating network is a feed-forward neural network with one hidden layer of size 250 and tanh non-linear activation function. The output layer is of size m, where m is the number of experts. Values of the output layer are normalized by applying softmax. The mixture model allows to back propagate errors both to the gating network and the experts themselves. However, considering the small size of the training data and the complexity of the experts, in terms of number of parameters, full back propagation is likely to result in over-fitting. Therefore, we only update the weights of the gating network, where the weights of the NMT predictors have been pre-trained separately. We train our mixture model for 10 epochs with AdaGrad with a learning rate of 0.001.

$$g = \mathit{softmax}(W_{\mathit{gate}} \mathit{tanh}(W_{\mathit{hid}}[f^1(x); ...; f^m(x)] + b_{\mathit{hid}}) + b_{\mathit{gate}}) \tag{10}$$

5 Translation experiments with trained ensembles

In the previous sections we have shown that NMT ensembles, and in particular multi-source ensembles, can improve translation quality. We proposed two methods to learn an ensemble combination function from data, which is more capable of exploiting the potential of ensembles than simple uniform weighting. In this section we test these methods in translation experiments. We compare the two ensemble induction methods (same-source systems with different parameter initializations and multi-source set) and apply all combination methods. All results are presented in Table 4.

[3] In our preliminary experiments, we also experimented with using other layers of the NMT model as the decoding state x: the top-most recurrent layer of decoder, the context vector, and the embedding of the previously predicted target word as the decoding state. However, using the non-recurrent hidden state $\widetilde{h}$ achieved the best results overall.

Ensemble set	Combination type					
	uniform		global		context-dependent	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
$de_1^2 \rightarrow$ en	21.8	50.4	21.8	50.4	21.8	50.3
$de_1^4 \rightarrow$ en	21.8	50.4	21.8	50.4	-	-
$\{de_1^2, de_3^4\} \rightarrow$ en	-	-	21.8	50.4	**22.8**	**51.0**
$fr_1^2 \rightarrow$ en	29.2	58.0	29.2	58.1	29.3	58.1
$fr_1^4 \rightarrow$ en	29.2	58.0	29.2	58.1	-	-
$\{fr_1^2, fr_3^4\} \rightarrow$ en	-	-	29.2	58.1	**30.2**	**59.0**
de,fr$\rightarrow$ en	**29.5**	**58.3**	**29.9**	**58.7**	30.3	59.2
de_1,de_2,fr_1,$fr_2 \rightarrow$ en	29.4	58.3	29.3	58.2	-	-
$\{\{de_1,fr_1\},\{de_2,fr_2\}\}\rightarrow$ en	-	-	29.2	57.9	**31.5**	**60.3**

Table 4: Translation experiments for the French, German into English scenario. *Ensemble set* designates ensemble induction method. *Combination type* refers to the method used to combine predictions during decoding. We use curly brackets to denote hierarchical ensemble combination.

For each ensemble set type, we evaluate ensembles of size 2 and 4. The notation below should be understood as follows: $de_k^{k+l} \rightarrow$ en stands for a single-source ensemble of l German-English systems. Analogously for French-English. de,fr $\rightarrow$ en is a multi-source ensemble of size 2, and we use subscript indices if there are more than 2 systems in an ensemble covering the same source language. We only apply context-dependent combination for ensembles of size 2 to avoid overfitting for bigger ensembles. Note that this does not prevent the application of our context-dependent combination method to bigger ensembles, as we can combine systems hierarchically. In a hierarchical ensemble the set of predictors is divided into disjoint subsets and each of the subsets is combined separately. The resulting combination systems can then be treated as predictors in a new ensemble, and thus can be further combined for a joint prediction. In our case the maximal number of predictors is 4, therefore our hierarchical ensembles have 2 levels. Hierarchical ensembles allow one to make prediction combinations multiple times which can further boost the potential of an ensemble. Since in this paper we only consider a low-resourced scenario with a small amount of training trilingual data, we do not train a hierarchical combination function. Instead, we do global or context dependent combination for ensemble sets at the bottom level (level of individual NMT systems) and then weight their outputs uniformly (level of combined systems). We use curly brackets denote hierarchical combination.

We see in Table 4 that multi-source ensembles generally perform better that single-source ensembles. The largest improvements for multi-source ensembles are due to our context-dependent combination method. This suggests that the trained gating network is able to capture linguistic context. We note that a multi-source ensemble with contextual combination outperforms the empirical upper bound estimated in Section 3, although it should be noted that this is an upper bound for a global combination method. On the other hand, for single-source ensembles, context-dependent combination (by itself) does not provide additional improvements as compared to global weighting. This suggests that the variation found in single-source ensembles is not as systematic as in multi-source ensembles. As part of future work, we are planning to perform a more linguistically oriented analysis to identify contexts triggering a high degree variation in ensembles. The results of such analysis will also provide the basis for a more linguistically oriented definition of the decoding state x as defined in Section 4.

We also note that simply increasing the size of an ensemble does not necessarily improve translation performance. Previous approaches using NMT ensembles often report performance increases for ensembles consisting of a larger number of systems, typically 8 or 12. One could therefore speculate that ensembles of 4 systems are not enough to significantly increase diversity as compared to an ensemble of size 2. Of course, our result are also influenced by several other factors such as the choice of languages, training data, etc. However, we should point out that hierarchically combining a set of 4 systems does improve translation quality. At this point, hierarchical combination still requires further investigation,

but for the time being, it can be seen as a simple 'recipe' to boost translation quality further.

6 Conclusions

In this paper we compared existing ensemble set induction methods for NMT and proposed a number of system combination methods: global (across instances) weighting of predictors' outputs and context-dependent weighting. Our main goal was to validate the linguistic hypothesis that translation systems from different source language into the same target language have complementary strengths and weaknesses in terms of translation performance and introduce an approach that can exploit the respective strenghts and weaknesses to achieve better translation quality. In our experiments with German-English and French-English we found that multi-source ensembles yield the best performance, compared to the individual translation systems, as well as compared to single-source ensembles of NMTs produced by different random initializations. This is an interesting finding because individually the two systems differ substantially in their translation quality. We also found that ensemble combination based on a gating network that decides how to combine systems at every prediction step achieves better performance as compared to a global (constant) combination function or uniform weighting in the majority of cases.

Overall this is a compelling result and it leaves us with a number of questions for future work. First, can we characterize linguistically what types of contexts are mores suited to be translated by a German-English system, and which are more suited to be translated by a French-English system? Gaining insights in that direction can help us answer another question: is there a better way to represent the current context which is the input to the gating network? In this paper we used a concatenation of each system's last hidden state $\widetilde{h}$, but a potentially more effective and linguistically more intuitive representation may be found. Finally, it would be interesting to see to what extent our approach can benefit from three or more mutually different source languages.

Acknowledgements

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project number 639.022.213 and a Google Faculty Research Award. We also thank NVIDIA for their hardware support. We thank Ke Tran for providing the neural machine translation baseline system and the anonymous reviewers for their helpful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. *to appear in Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, October. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54. Association for Computational Linguistics.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *Association for Computational Lingustics*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *European Chapter of the Association for Computational Linguistics*.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of MT Summit*, volume 8, pages 253–258.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-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*, pages 311–318. Association for Computational Linguistics.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 719–727. Association for Computational Linguistics.

Lane Schwartz. 2008. Multi-source translation methods. In *Proceedings of AMTA 2008*, October.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. *Proceedings of the First Conference on Machine Translation, Volume 1: Research Papers. Berlin, Germany, pp. 83-91.*

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.