Translation Modeling with Bidirectional Recurrent Neural Networks

Martin Sundermeyer[1], **Tamer Alkhouli**[1], **Joern Wuebker**[1], **and Hermann Ney**[1,2]
[1]Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
[2]Spoken Language Processing Group
Univ. Paris-Sud, France and LIMSI/CNRS, Orsay, France
{surname}@cs.rwth-aachen.de

Abstract

This work presents two different translation models using recurrent neural networks. The first one is a word-based approach using word alignments. Second, we present phrase-based translation models that are more consistent with phrase-based decoding. Moreover, we introduce bidirectional recurrent neural models to the problem of machine translation, allowing us to use the full source sentence in our models, which is also of theoretical interest. We demonstrate that our translation models are capable of improving strong baselines already including recurrent neural language models on three tasks: IWSLT 2013 German→English, BOLT Arabic→English and Chinese→English. We obtain gains up to 1.6% BLEU and 1.7% TER by rescoring 1000-best lists.

1 Introduction

Neural network models have recently experienced unprecedented attention in research on statistical machine translation (SMT). Several groups have reported strong improvements over state-of-the-art baselines using feedforward neural network-based language models (Schwenk et al., 2006; Vaswani et al., 2013), as well as translation models (Le et al., 2012; Schwenk, 2012; Devlin et al., 2014). Different from the feedforward design, *recurrent* neural networks (RNNs) have the advantage of being able to take into account an unbounded history of previous observations. In theory, this enables them to model long-distance dependencies of arbitrary length. However, while previous work on translation modeling with recurrent neural networks shows its effectiveness on standard baselines, so far no notable gains have been presented on top of recurrent language models (Auli et al., 2013; Kalchbrenner and Blunsom, 2013; Hu et al., 2014).

In this work, we present two novel approaches to recurrent neural translation modeling: word-based and phrase-based. The word-based approach assumes one-to-one aligned source and target sentences. We evaluate different ways of resolving alignment ambiguities to obtain such alignments. The phrase-based RNN approach is more closely tied to the underlying translation paradigm. It models actual phrasal translation probabilities while avoiding sparsity issues by using single words as input and output units. Furthermore, in addition to the unidirectional formulation, we are the first to propose a bidirectional architecture which can take the full source sentence into account for all predictions. Our experiments show that these models can improve state-of-the-art baselines containing a recurrent language model on three tasks. For our competitive IWSLT 2013 German→English system, we observe gains of up to 1.6% BLEU and 1.7% TER. Improvements are also demonstrated on top of our evaluation systems for BOLT Arabic→English and Chinese→English, which also include recurrent neural language models.

The rest of this paper is structured as follows. In Section 2 we review related work and in Section 3 an overview of long short-term memory (LSTM) neural networks, a special type of recurrent neural networks we make use of in this work, is given. Section 4 describes our novel translation models. Finally, experiments are presented in Section 5 and we conclude with Section 6.

2 Related Work

In this Section we contrast previous work to ours, where we design RNNs to model bilingual dependencies, which are applied to rerank n-best lists after decoding.

To the best of our knowledge, the earliest attempts to apply neural networks in machine translation (MT) are presented in (Castaño et al., 1997; Castaño and Casacuberta, 1997; Castaño and Casacuberta, 1999), where they were used for example-based MT.

Recently, Le et al. (2012) presented translation models using an output layer with classes and a shortlist for rescoring using feedforward networks. They compare between word-factored and tuple-factored n-gram models, obtaining their best results using the word-factored approach, which is less amenable to data sparsity issues. Both of our word-based and phrase-based models eventually work on the word level. Kalchbrenner and Blunsom (2013) use recurrent neural networks with full source sentence representations. The continuous representations are obtained by applying a sequence of convolutions, and the result is fed into the hidden layer of a recurrent language model. Rescoring results indicate no improvements over the state of the art. Auli et al. (2013) also include source sentence representations built either using Latent Semantic Analysis or by concatenating word embeddings. This approach produced no notable gain over systems using a recurrent language model. On the other hand, our proposed bidirectional models include the full source sentence relying on recurrency, yielding improvements over competitive baselines already including a recurrent language model.

RNNs were also used with minimum translation units (Hu et al., 2014), which are phrase pairs undergoing certain constraints. At the input layer, each of the source and target phrases are modeled as a bag of words, while the output phrase is predicted word-by-word assuming conditional independence. The approach seeks to alleviate data sparsity problems that would arise if phrases were to be uniquely distinguished. Our proposed phrase-based models maintain word order within phrases, but the phrases are processed in a word-pair manner, while the phrase boundaries remain implicitly encoded in the way the words are presented to the network. Schwenk (2012) proposed a feedforward network that predicts phrases of a fixed maximum length, such that all phrase words are predicted at once. The prediction is conditioned on the source phrase. Since our phrase-based model predicts one word at a time, it does not assume any phrase length. Moreover, our model's predictions go beyond phrase boundaries and cover unbounded history and future contexts.

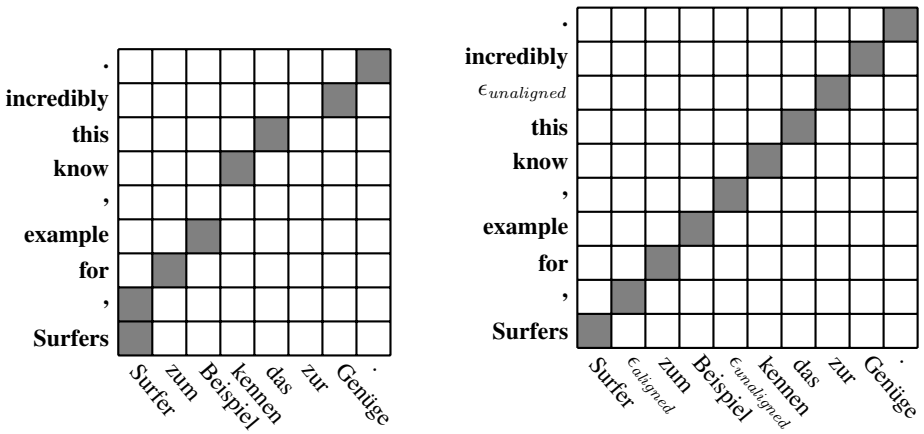
Using neural networks during decoding requires tackling the costly output normalization step. Vaswani et al. (2013) avoid this step by training feedforward neural language models using *noise contrastive estimation*, while Devlin et al. (2014) augment the training objective function to produce approximately normalized scores directly. The latter work makes use of translation and joint models, and pre-computes the first hidden layer beforehand, resulting in large speedups. They report major improvements over strong baselines. The speedups achieved by both works allowed to integrate feedforward neural networks into the decoder.

3 LSTM Recurrent Neural Networks

Our work is based on recurrent neural networks. In related fields like e. g. language modeling, this type of neural network has been shown to perform considerably better than standard feedforward architectures (Mikolov et al., 2011; Arisoy et al., 2012; Sundermeyer et al., 2013; Liu et al., 2014).

Most commonly, recurrent neural networks are trained with stochastic gradient descent (SGD), where the gradient of the training criterion is computed with the backpropagation through time algorithm (Rumelhart et al., 1986; Werbos, 1990; Williams and Zipser, 1995). However, the combination of RNN networks with conventional backpropagation training leads to conceptual difficulties which are known as the vanishing (or exploding) gradient problem, described e. g. in (Bengio et al., 1994). To remedy this problem, in (Hochreiter and Schmidhuber, 1997) it was suggested to modify the architecture of a standard RNN in such a way that vanishing and exploding gradients are avoided during backpropagation. In particular, no modification of the training algorithm is necessary. The resulting architecture is referred to as long short-term memory (LSTM) neural network.

Bidirectional recurrent neural networks (BRNNs) were first proposed in (Schuster and Paliwal, 1997) and applied to speech recognition tasks. They have been since applied to different



(a) Original

(b) One-to-one alignment

Figure 1: Example sentence from the German→English IWSLT data. The one-to-one alignment is created by introducing $\epsilon_{aligned}$ and $\epsilon_{unaligned}$ tokens.

tasks like parsing (Henderson, 2004) and spoken language understanding (Mesnil et al., 2013). Bidirectional long short-term memory (BLSTM) networks are BRNNs using LSTM hidden layers (Graves and Schmidhuber, 2005). This work introduces BLSTMs to the problem of machine translation, allowing powerful models that employ unlimited history and future information to make predictions.

While the proposed models do not make any assumptions about the type of RNN used, all of our experiments make use of recurrent LSTM neural networks, where we include later LSTM extensions proposed in (Gers et al., 2000; Gers et al., 2003). The cross-entropy error criterion is used for training. Further details on LSTM neural networks can be found in (Graves and Schmidhuber, 2005; Sundermeyer et al., 2012).

4 Translation Modeling with RNNs

In the following we describe our word- and phrase-based translation models in detail. We also show how bidirectional RNNs can enable such models to include full source information.

4.1 Resolving Alignment Ambiguities

Our word-based recurrent models are only defined for one-to-one-aligned source-target sentence pairs. In this work, we always evaluate the model in the order of the target sentence. However, we experiment with several different ways to resolve ambiguities due to unaligned or multiply aligned words. To that end, we introduce two additional tokens, $\epsilon_{aligned}$ and $\epsilon_{unaligned}$. Unaligned words are either removed or aligned to an extra $\epsilon_{unaligned}$ token on the opposite side. If an $\epsilon_{unaligned}$ is introduced on the target side, its position is determined by the aligned source word that is closest to the unaligned source word in question, preferring left to right. To resolve one-to-many alignments, we use an IBM-1 translation table to decide for one of the alignment connections to be kept. The remaining words are also either deleted or aligned to additionally introduced $\epsilon_{aligned}$ tokens on the opposite side. Fig. 1 shows an example sentence from the IWSLT data, where all ϵ tokens are introduced.

In a short experiment, we evaluated 5 different setups with our unidirectional RNN translation model (cf. next Section): without any ϵ tokens, without $\epsilon_{unaligned}$, source identity, target identity and using all ϵ tokens. Source identity means we

	dev		test	
	BLEU	TER	BLEU	TER
baseline	33.5	45.8	30.9	48.4
w/o ϵ	34.2	45.3	31.8	47.7
w/o $\epsilon_{unaligned}$	34.4	44.8	31.7	47.4
source identity	34.5	45.0	31.9	47.5
target identity	34.5	44.6	31.9	47.0
all ϵ	34.6	44.5	32.0	47.1

Table 1: Comparison of including different sets of ϵ tokens into the one-to-one alignment on the IWSLT 2013 German→English task using the unidirectional RNN translation model.

introduce no ϵ tokens on source side, but all on target side. Target identity is defined analogously. The results can be found in Tab. 1. We use the setup with all ϵ tokens in all following experiments, which showed the best BLEU performance.

4.2 Word-based RNN Models

Given a pair of source sequence $f_1^I = f_1 \dots f_I$ and target sequence $e_1^I = e_1 \dots e_I$, where we assume a direct correspondence between f_i and e_i, we define the posterior translation probability by factorizing on the target words:

$$p(e_1^I|f_1^I) = \prod_{i=1}^{I} p(e_i|e_1^{i-1}, f_1^I) \quad (1)$$

$$\approx \prod_{i=1}^{I} p(e_i|e_1^{i-1}, f_1^{i+d}) \quad (2)$$

$$\approx \prod_{i=1}^{I} p(e_i|f_1^{i+d}). \quad (3)$$

We denote the formulation (1) as the bidirectional joint model (BJM). This model can be simplified by several independence assumptions. First, we drop the dependency on the future source information, receiving what we denote as the unidirectional joint model (JM) in (2). Here, $d \in \mathbb{N}_0$ is a delay parameter, which is set to $d = 0$ for all experiments, except for the comparative results reported in Fig. 7. Finally, assuming conditional independence from the previous target sequence, we receive the unidirectional translation model (TM) in (3). Analogously, we can define a bidirectional translation model (BTM) by keeping the dependency on the full source sentence f_1^I, but dropping the previous target sequence e_1^{i-1}:

$$p(e_1^I|f_1^I) \approx \prod_{i=1}^{I} p(e_i|f_1^I). \quad (4)$$

Fig. 2 shows the dependencies of the word-based neural translation and joint models. The alignment points are traversed in target order and at each time step one target word is predicted. The pure translation model (TM) takes only source words as input, while the joint model (JM) takes the preceding target words as an additional input. A delay of $d > 0$ is implemented by shifting the target sequence by d time steps and filling the first d target positions and the last d source positions with a dedicated $\epsilon_{padding}$ symbol. The RNN architecture for the unidirectional word-based models

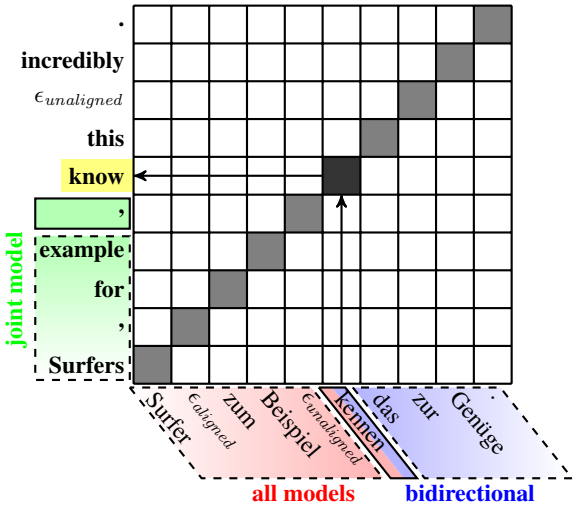


Figure 2: Dependencies modeled within the word-based RNN models when predicting the target word 'know'. Directly processed information is depicted with solid rectangles, and information available through recurrent connections is marked with dashed rectangles.

is illustrated in Fig. 3, which corresponds to the following set of equations:

$$\begin{aligned} y_i &= A_1 \hat{f}_i + A_2 \hat{e}_{i-1} \\ z_i &= \xi(y_i; A_3, y_1^{i-1}) \\ p\big(c(e_i)|e_1^{i-1}, f_1^i\big) &= \varphi_{c(e_i)}(A_4 z_i) \\ p\big(e_i|c(e_i), e_1^{i-1}, f_1^i\big) &= \varphi_{e_i}(A_{c(e_i)} z_i) \\ p(e_i|e_1^{i-1}, f_1^i) &= p\big(e_i|c(e_i), e_1^{i-1}, f_1^i\big) \cdot \\ &\quad p\big(c(e_i)|e_1^{i-1}, f_1^i\big) \end{aligned}$$

Here, by $\hat{f}_i$ and $\hat{e}_{i-1}$ we denote the one-hot encoded vector representations of the source and target words f_i and e_{i-1}. The outgoing activation values of the projection layer and the LSTM layer are y_i and z_i, respectively. The matrices A_j contain the weights of the neural network layers. By $\xi(\cdot\,; A_3, y_1^{i-1})$ we denote the LSTM formalism that we plug in at the third layer. As the LSTM layer is recurrent, we explicitly include the dependence on the previous layer activations y_1^{i-1}. Finally, φ is the widely-used softmax function to obtain normalized probabilities, and c denotes a word class mapping from any target word to its unique word class. For the bidirectional model, the equations can be defined analogously.

Due to the use of word classes, the output layer consists of two parts. The class probability $p\big(c(e_i)|e_1^{i-1}, f_1^i\big)$ is computed first, and then

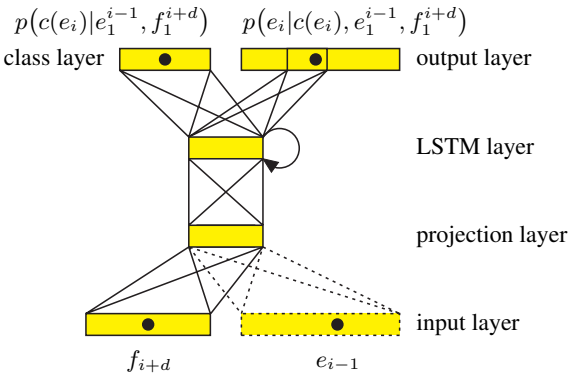


Figure 3: Architecture of a recurrent unidirectional translation model. By including the dashed parts, a *joint* model is obtained.

the word probability $p(e_i|c(e_i), e_1^{i-1}, f_1^i)$ is obtained given the word class. This trick helps avoiding the otherwise computationally expensive normalization sum, which would be carried out over all words in the target vocabulary. In a class-factorized output layer where each word belongs to a single class, the normalization is carried out over all classes, whose number is typically much less than the vocabulary size. The other normalization sum needed to produce the word probability is limited to the words belonging to the same class (Goodman, 2001; Morin and Bengio, 2005).

4.3 Phrase-based RNN Models

One of the conceptual disadvantages of word-based modeling as introduced in the previous section is that there is a mismatch between training and testing conditions: During neural network training, the vocabulary has to be extended by additional ϵ tokens, and a one-to-one alignment is used which does not reflect the situation in decoding. In phrase-based machine translation, more complex alignments in terms of multiple words on both the source and the target sides are used, which allow the decoder to make use of richer short-distance dependencies and are crucial for the performance of the resulting system.

From this perspective, it seems interesting to standardize the alignments used in decoding, and in training the neural network. However, it is difficult to use the phrases themselves as the vocabulary of the RNN. Usually, the huge number of potential phrases in comparison to the relatively small amount of training data makes the learning of continuous phrase representations difficult

Surfers Surfer	, for example , zum Beispiel	know kennen	this das	incredibly zur Genüge	. .

Figure 4: Example phrase alignment for a sentence from the IWSLT training data.

due to data sparsity. This is confirmed by results presented in (Le et al., 2012), which show that a word-factored translation model outperforms the phrase-factored version. Therefore, in this work we continue relying on source and target word vocabularies for building our phrase representations. However, we no longer use a direct correspondence between a source and a target word, as enforced in our word-based models.

Fig. 4 shows an example phrase alignment, where a sequence of source words $\tilde{f}_i$ is directly mapped to a sequence of target words $\tilde{e}_i$ for $1 \leq i \leq \tilde{I}$. By $\tilde{I}$, we denote the number of phrases in the alignment. We decompose the target sentence posterior probability in the following way:

$$p(e_1^I|f_1^J) = \prod_{i=1}^{\tilde{I}} p(\tilde{e}_i|\tilde{e}_1^{i-1}, \tilde{f}_1^{\tilde{I}}) \quad (5)$$

$$\approx \prod_{i=1}^{\tilde{I}} p(\tilde{e}_i|\tilde{e}_1^{i-1}, \tilde{f}_1^{i}) \quad (6)$$

where the joint model in Eq. 5 would correspond to a bidirectional RNN, and Eq. 6 only requires a unidirectional RNN. By leaving out the conditioning on the target side, we obtain a phrase-based *translation* model.

As there is no one-to-one correspondence between the words within a phrase, the basic idea of our phrase-based approach is to let the neural network learn the dependencies itself, and present the full source side of the phrase to the network before letting it predict target side words. Then the probability for the target side of a phrase can be computed, in case of Eq. 6, by:

$$p(\tilde{e}_i|\tilde{e}_1^{i-1}, \tilde{f}_1^{\tilde{I}}) = \prod_{j=1}^{|\tilde{e}_i|} p\big((\tilde{e}_i)_j|(\tilde{e}_i)_1^{j-1}, \tilde{e}_1^{i-1}, \tilde{f}_1^{i}\big),$$

and analogously for the case of Eq. 5. Here, $(\tilde{e}_i)_j$ denotes the j-th word of the i-th aligned target phrase.

We feed the source side of a phrase into the neural network one word at a time. Only when the

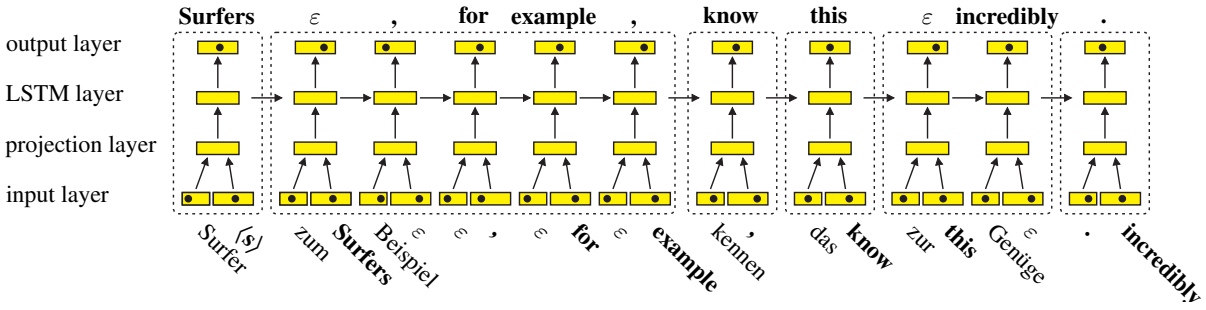


Figure 5: A recurrent phrase-based joint translation model, unfolded over time. Source words are printed in normal face, while target words are printed in bold face. Dashed lines indicate phrases from the example sentence. For brevity, we omit the precise handling of sentence begin and end tokens.

presentation of the source side is finished we start estimating probabilities for the target side. Therefore, we do not let the neural network learn a target distribution until the very last source word is considered. In this way, we break up the conventional RNN training scheme where an input sample is directly followed by its corresponding teacher signal. Similarly, the presentation of the source side of the next phrase only starts after the prediction of the current target side is completed.

To this end, we introduce a no-operation token, denoted by ε, which is not part of the vocabulary (which means it cannot be input to or predicted by the RNN). When the ε token occurs as input, it indicates that no input needs to be processed by the RNN. When the ε token occurs as a teacher signal for the RNN, the output layer distribution is ignored, and does not even have to be computed. In both cases, all the other layers are still processed during forward and backward passes such that the RNN state can be advanced even without additional input or output.

Fig. 5 depicts the evaluation of a phrase-based joint model for the example alignment from Fig. 4. For a source phrase $\tilde{f}_i$, we include $(|\tilde{e}_i|-1)$ many ε symbols at the end of the phrase. Conversely, for a target phrase $\tilde{e}_i$, we include $(|\tilde{f}_i| - 1)$ many ε symbols at the beginning of the phrase.

E. g., in the figure, the second dashed rectangle from the left depicts the training of the English phrase "*, for example ,*" and its German translation "*zum Beispiel*". At the input layer, we feed in the source words one at a time, while we present ε tokens at the target side input layer and the output layer (with the exception of the very first time step, where we still have the last target word from the previous phrase as input instead of ε). With the last word of the source phrase "*Beispiel*" being presented to the network, the full source phrase is stored in the hidden layer, and the neural network is then trained to predict the target phrase words at the output layer. Subsequently, the source input is ε, and the target input is the most recent target side history word.

To obtain a phrase-aligned training sequence for the phrase-based RNN models, we force-align the training data with the application of leave-one-out as described in (Wuebker et al., 2010).

4.4 Bidirectional RNN Architecture

While the unidirectional RNNs include an unbounded sentence history, they are still limited in the number of future source words they include. Bidirectional models provide a flexible means to also include an unbounded future context, which, unlike the delayed unidirectional models, require no tuning to determine the amount of delay.

Fig. 6 illustrates the bidirectional model architecture, which is an extension of the unidirectional model of Fig. 3. First, an additional recurrent hidden layer is added in parallel to the existing one. This layer will be referred to as the backward layer, since it processes information in backward time direction. This hidden layer receives source word input only, while target words in the case of a joint model are fed to the forward layer as in the unidirectional case. Due to the backward recurrency, the backward layer will make the information f_i^I available when predicting the target word e_i, while the forward layer takes care of the source history f_1^i. Jointly, the forward and backward branches include the full source sentence f_1^I, as indicated in Fig. 2. Fig. 6 shows the "deep" variant of the bidirectional model, where the for-

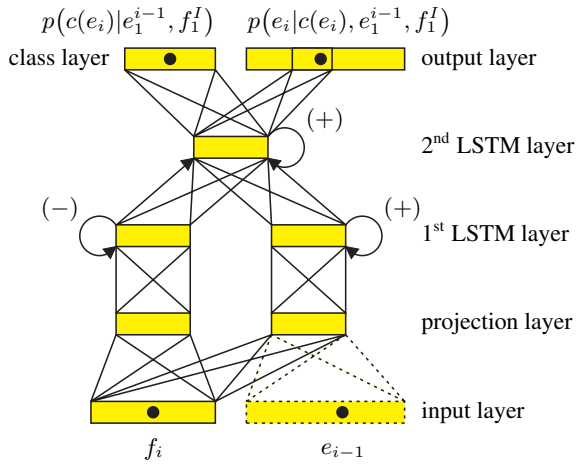


Figure 6: Architecture of a recurrent bidirectional translation model. By $(+)$ and $(-)$, we indicate a processing in forward and backward time directions, respectively. The inclusion of the dashed parts leads to a bidirectional *joint* model. One source projection matrix is used for the forward and backward branches.

ward and backward layers converge into a hidden layer. A shallow variant can be obtained if the parallel layers converge into the output layer directly[1].

Due to the full dependence on the source sequence, evaluating bidirectional networks requires computing the forward pass of the forward and backward layers for the full sequence, before being able to evaluate the next layers. In the backward pass of backpropagation, the forward and backward recurrent layers are processed in decreasing and increasing time order, respectively.

5 Experiments

5.1 Setup

All translation experiments are performed with the *Jane* toolkit (Vilar et al., 2010; Wuebker et al., 2012). The largest part of our experiments is carried out on the IWSLT 2013 German→English shared translation task.[2] The baseline system is trained on all available bilingual data, 4.3M sentence pairs in total, and uses a 4-gram LM with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998), trained with the SRILM toolkit (Stolcke, 2002). As additional data sources for the LM we selected parts of the Shuffled News and LDC English Gigaword corpora based on cross-entropy difference (Moore and Lewis, 2010), resulting in a total of 1.7 billion running words for LM training. The state-of-the-art baseline is a standard phrase-based SMT system (Koehn et al., 2003) tuned with MERT (Och, 2003). It contains a hierarchical reordering model (Galley and Manning, 2008) and a 7-gram word cluster language model (Wuebker et al., 2013). Here, we also compare against a feed-forward joint model as described by Devlin et al. (2014), with a source window of 11 words and a target history of three words, which we denote as *BBN-JM*. Instead of POS tags, we predict word classes trained with `mkcls`. We use a shortlist of size 16K and 1000 classes for the remaining words. All neural networks are trained on the TED portion of the data (138K segments) and are applied in a rescoring step on 1000-best lists.

To confirm our results, we run additional experiments on the Arabic→English and Chinese→English tasks of the DARPA BOLT project. In both cases, the neural network models are added on top of our most competitive evaluation system. On Chinese→English, we use a hierarchical phrase-based system trained on 3.7M segments with 22 dense features, including an advanced orientation model (Huck et al., 2013). For the neural network training, we selected a subset of 9M running words. The Arabic→English system is a standard phrase-based decoder trained on 6.6M segments, using 17 dense features. The neural network training was performed using a selection amounting to 15.5M running words. For both tasks we apply the neural networks by rescoring 1000-best lists and evaluate results on two data sets from the 'discussion forum' domain, `test1` and `test2`. The sizes of the data sets for the Arabic→English system are: 1219 (`dev`), 1510 (`test1`), and 1137 (`test2`) segments, and for the Chinese→English system are: 5074 (`dev`), 1844 (`test1`), and 1124 (`test2`) segments. All results are measured in case-insensitive BLEU [%] (Papineni et al., 2002) and TER [%] (Snover et al., 2006) on a single reference.

5.2 Results

Our results on the IWSLT German→English task are summarized in Tab. 2. At this point, we do not include a recurrent neural network lan-

[1] In our implementation, the forward and backward layers converge into an intermediate identity layer, and the aggregate is weighted and fed to the next layer.

[2] `http://www.iwslt2013.org`

	dev		test	
	BLEU	TER	BLEU	TER
baseline	33.5	45.8	30.9	48.4
TM	34.6	44.5	32.0	47.1
JM	34.7	44.7	31.8	47.4
BTM	34.7	44.9	32.3	47.0
BTM (deep)	34.8	44.3	**32.5**	46.7
BJM	34.7	44.5	32.1	47.0
BJM (deep)	**34.9**	**44.1**	32.2	**46.6**
PTM	34.3	44.9	32.1	47.5
PJM	34.3	45.0	32.0	47.5
PJM (10-best)	34.4	44.8	32.0	47.3
PJM (deep)	34.6	44.7	32.0	47.6
PBJM (deep)	34.8	44.9	31.9	47.5
BBN-JM	34.4	44.9	31.9	47.6

Table 2: Results for the IWSLT 2013 German→English task with different RNN models. T: translation, J: joint, B: bidirectional, P: phrase-based.

guage model yet. Here, the delay parameter d from Equations 2 and 3 is set to zero. We observe that for all recurrent translation models, we achieve substantial improvements over the baseline on the test data, ranging from 0.9 BLEU up to 1.6 BLEU. These results are also consistent with the improvements in terms of TER, where we achieve reductions by 0.8 TER up to 1.8 TER.

These numbers can be directly compared to the case of feedforward neural network-based translation modeling as proposed in (Devlin et al., 2014) which we include in the very last row of the table. Nearly all of our recurrent models outperform the feedforward approach, where the RNN model performing best on the dev data is better on test by 0.3 BLEU and 1.0 TER.

Interestingly, for the recurrent word-based models, on the test data it can be seen that TMs perform better than JMs, even though TMs do not take advantage of the target side history words. However, exploiting this extra information does not always need to result in a better model, as the target side words are only derived from the given source side, which is available to both TMs and JMs. On the other hand, including future source words in a bidirectional model clearly improves the performance further. By adding another LSTM

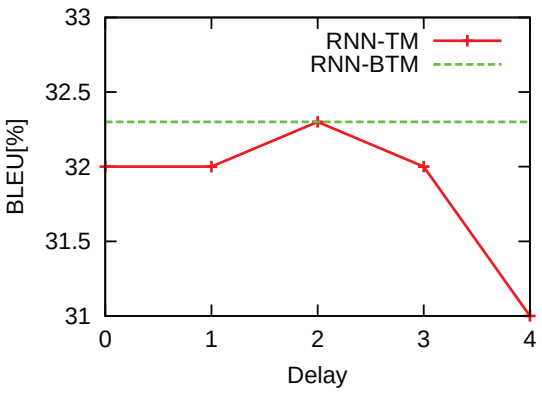


Figure 7: BLEU scores on the IWSLT test set with different delays for the unidirectional RNN-TM and the bidirectional RNN-BTM.

layer that combines forward and backward time directions (indicated as 'deep' in the table), we obtain our overall best model.

In Fig. 7 we compare the word-based bidirectional TM with a unidirectional TM that uses different time delays $d = 0, \ldots, 4$. For a delay $d = 2$, the same performance is obtained as with the bidirectional model, but this comes at the price of tuning the delay parameter.

In comparison to the unidirectional word-based models, phrase-based models perform similarly. In the tables, we include those phrase-based variants which perform best on the dev data, where phrase-based JMs always are at least as good or better than the corresponding TMs in terms of BLEU. Therefore, we mainly report JM results for the phrase-based networks. A phrase-based model can also be trained on multiple variants for the phrase alignment. For our experiments, we tested 10-best alignments against the single best alignment, which resulted in a small improvement of 0.2 TER on both dev and test. We did not observe consistent gains by using an additional hidden layer or bidirectional models. To some extent, future information is already considered in unidirectional phrase-based models by feeding the complete source side before predicting the target side.

Tab. 3 shows different model combination results for the IWSLT task, where a recurrent language model is included in the baseline. Adding a deep bidirectional TM or JM to the recurrent language model improves the RNN-LM baseline by 1.2 BLEU or 1.1 BLEU, respectively. A phrase-based model substantially improves over

	dev		eval11		test	
	BLEU[%]	TER[%]	BLEU[%]	TER[%]	BLEU[%]	TER[%]
baseline (w/ RNN-LM)	34.3	44.8	36.4	42.9	31.5	47.8
BTM (deep)	34.9	**43.7**	**37.6**	**41.5**	**32.7**	**46.1**
BJM (deep)	**35.0**	44.4	37.4	41.9	32.6	46.5
PBJM (deep)	34.8	44.6	36.9	42.6	32.3	47.2
4 RNN models	**35.2**	**43.4**	**38.0**	**41.2**	**32.7**	**46.0**

Table 3: Results for the IWSLT 2013 German→English task with different RNN models. All results include a recurrent language model. T: translation, J: joint, B: bidirectional, P: phrase-based.

the RNN-LM baseline, but performs not as good as its word-based counterparts. By adding four different translation models, including models in reverse word order and reverse translation direction, we are able to improve these numbers even further. However, especially on the test data, the gains from model combination saturate quickly.

Apart from the IWSLT track, we also analyze the performance of our translation models on the BOLT Chinese→English and Arabic→English translation tasks. Due to the large amount of training data, we concentrate on models of high performance in the IWSLT experiments. The results can be found in Tab. 4 and 5. In both cases, we see consistent improvements over the recurrent neural network language model baseline, improving the Arabic→English system by 0.6 BLEU and 0.5 TER on test1. This can be compared to the rescoring results for the same task reported by (Devlin et al., 2014), where they achieved 0.3 BLEU, despite the fact that they used multiple references for scoring, whereas in our experiments we rely on a single reference only. The models are also able to improve the Chinese→English system by 0.5 BLEU and 0.5 TER on test2.

	test1		test2	
	BLEU	TER	BLEU	TER
baseline	25.2	57.4	26.8	57.3
BTM (deep)	25.6	**56.6**	26.8	56.7
BJM (deep)	25.9	56.9	27.4	56.7
RNN-LM	25.6	57.1	27.5	56.7
+ BTM (deep)	25.9	56.7	27.3	56.8
+ BJM (deep)	**26.2**	**56.6**	**27.9**	**56.5**

Table 4: Results for the BOLT Arabic→English task with different RNN models. The "+" sign in the last two rows indicates that either of the corresponding deep models (BTM and BJM) are added to the baseline including the recurrent language model (i.e. they are not applied at the same time). T: translation, J: joint, B: bidirectional.

5.3 Analysis

To investigate whether bidirectional models benefit from future source information, we compare the single-best output of a system reranked with a unidirectional model to the output reranked with a bidirectional model. We choose the models to be translation models in both cases, as they predict target words independent of previous predictions, given the source information (cf. Eqs. (3, 4)). This makes it easier to detect the effect of including future source information or the lack thereof. The examples are taken from the IWSLT task, where we include the one-to-one source information, reordered according to the target side.

source: nicht **so wie** ich
reference: not **like** me
Hypothesis 1:
1-to-1 source: **so** ich ϵ nicht **wie**
1-to-1 target: **so** I do n't **like**
Hypothesis 2:
1-to-1 source: nicht **so wie** ich
1-to-1 target: not ϵ **like** me

In this example, the German phrase "*so wie*" translates to "*like*" in English. The bidirectional model prefers hypothesis 2, making use of the future word "*wie*" when translating the German word "*so*" to ϵ, because it has future insight that this move will pay off later when translating

	BLEU	TER	BLEU	TER
baseline	18.3	63.6	16.7	63.0
BTM (deep)	18.7	63.3	17.1	62.6
BJM (deep)	18.5	**63.1**	17.2	**62.3**
RNN-LM	18.8	63.3	17.2	62.8
+ BTM (deep)	**18.9**	**63.1**	**17.7**	**62.3**
+ BJM (deep)	18.8	63.3	17.5	62.5

Table 5: Results for the BOLT Chinese→English task with different RNN models. The "+" sign in the last two rows indicates that either of the corresponding deep models (BTM and BJM) are added to the baseline including the recurrent language model (i.e. they are not applied at the same time). T: translation, B: bidirectional.

the rest of the sentence. This information is not available to the unidirectional model, which prefers hypothesis 1 instead.

> *source*: das taten wir dann auch und verschafften uns so **eine Zeit lang** einen Wettbewerbs Vorteil .
> *reference*: and we actually did that and it gave us a competitive advantage **for a while** .
> **Hypothesis 1**:
> *1-to-1 source*: das ϵ ϵ ϵ wir dann auch taten und verschafften uns so **eine Zeit lang** einen Wettbewerbs Vorteil .
> *1-to-1 target*: that 's just what we ϵ ϵ did and gave us ϵ a **time ,** a competitive advantage .
> **Hypothesis 2**:
> *1-to-1 source*: das ϵ ϵ ϵ wir dann auch taten und verschafften uns so einen Wettbewerbs Vorteil ϵ **eine Zeit lang** .
> *1-to-1 target*: that 's just what we ϵ ϵ did and gave us ϵ a competitive advantage **for a ϵ while** .

Here, the German phrase "*eine Zeit lang*" translates to "*for a while*" in English. Bidirectional scoring favors hypothesis 2, while unidirectional scoring favors hypothesis 1. It seems that the unidirectional model translates "*Zeit*" to "*time*" as the object of the verb "*give*" in hypothesis 1, being blind to the remaining part "*lang*" of the phrase which changes the meaning. The bidirectional model, to its advantage, has the full source information, allowing it to make the correct prediction.

6 Conclusion

We developed word- and phrase-based RNN translation models. The former is simple and performs well in practice, while the latter is more consistent with the phrase-based paradigm. The approach inherently evades data sparsity problems as it works on words in its lowest level of processing. Our experiments show the models are able to achieve notable improvements over baselines containing a recurrent LM.

In addition, and for the first time in statistical machine translation, we proposed a bidirectional neural architecture that allows modeling past and future dependencies of any length. Besides its good performance in practice, the bidirectional architecture is of theoretical interest as it allows the exact modeling of posterior probabilities.

Acknowledgments

This material is partially based upon work supported by the DARPA BOLT project under Contract No. HR0011- 12-C-0015. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements n^o 287658 and n^o 287755. Experiments were performed with computing resources granted by JARA-HPC from RWTH Aachen University under project 'jara0085'. We would like to thank Jan-Thorsten Peter for providing the *BBN-JM* system.

References

Ebru Arisoy, Tara N. Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. 2012. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28. Association for Computational Linguistics.

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint Language and Translation Modeling with Recurrent Neural Networks. In *Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054, Seattle, USA, October.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gra-

dient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

M. Asunción Castaño and Francisco Casacuberta. 1997. A connectionist approach to machine translation. In *5th International Conference on Speech Communication and Technology (EUROSPEECH-97)*, Rhodes, Greece.

M. Asunción Castaño and Francisco Casacuberta. 1999. Text-to-text machine translation using the RECONTRA connectionist model. In *Lecture Notes in Computer Science (IWANN 99)*, volume 1607, pages 683–692, Alicante, Spain.

M. Asunción Castaño, Francisco Casacuberta, and Enrique Vidal. 1997. Machine translation using neural networks and finite-state models. In *7th International Conference on Theoretical and Methodological Issues in Machine Translation. TMI'97*, pages 160–167, Santa Fe, USA.

Stanley F. Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, August.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *52nd Annual Meeting of the Association for Computational Linguistics*, page to appear, Baltimore, MD, USA, June.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856, Stroudsburg, PA, USA. Association for Computational Linguistics.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. 2003. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143.

Joshua Goodman. 2001. Classes for fast maximum entropy training. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 561–564. IEEE.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610.

James Henderson. 2004. Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 95. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yuening Hu, Michael Auli, Qin Gao, and Jianfeng Gao. 2014. Minimum translation modeling with recurrent neural networks. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 20–29, Gothenburg, Sweden, April. Association for Computational Linguistics.

Matthias Huck, Joern Wuebker, Felix Rietig, and Hermann Ney. 2013. A phrase orientation model for hierarchical machine translation. In *ACL 2013 Eighth Workshop on Statistical Machine Translation*, pages 452–463, Sofia, Bulgaria, August.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processingw*, volume 1, pages 181–184, May.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 127–133, Edmonton, Alberta.

Hai Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous Space Translation Models with Neural Networks. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–48, Montreal, Canada, June.

X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland. 2014. Efficient lattice rescoring using recurrent neural network language models. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4941–4945. IEEE.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

R.C. Moore and W. Lewis. 2010. Intelligent Selection of Language Model Training Data. In *ACL (Short Papers)*, pages 220–224, Uppsala, Sweden, July.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. *Learning Internal Representations by Error Propagation. In: J. L. McClelland, D. E. Rumelhart, and The PDP Research Group: "Parallel Distributed Processing, Volume 1: Foundations"*. The MIT Press.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. 2006. Continuous Space Language Models for Statistical Machine Translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730, Sydney, Australia, July.

Holger Schwenk. 2012. Continuous Space Translation Models for Phrase-Based Statistical Machine Translation. In *25th International Conference on Computational Linguistics (COLING)*, pages 1071–1080, Mumbai, India, December.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of the Int. Conf. on Speech and Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO, September.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Interspeech*, Portland, OR, USA, September.

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

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA, October. Association for Computational Linguistics.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 262–270, Uppsala, Sweden, July.

Paul J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

R. J. Williams and D. Zipser. 1995. *Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity. In: Yves Chauvain and David E. Rumelhart: "Back-Propagation: Theory, Architectures and Applications"*. Lawrence Erlbaum Publishers.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, pages 475–484, Uppsala, Sweden, July.

Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012. Jane 2: Open source phrase-based and hierarchical statistical machine translation. In *International Conference on Computational Linguistics*, pages 483–491, Mumbai, India, December.

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, USA, October.