Unsupervised Pretraining for Sequence to Sequence Learning

Prajit Ramachandran and **Peter J. Liu** and **Quoc V. Le**
Google Brain
{prajit, peterjliu, qvl}@google.com

Abstract

This work presents a general unsupervised learning method to improve the accuracy of sequence to sequence (seq2seq) models. In our method, the weights of the encoder and decoder of a seq2seq model are initialized with the pretrained weights of two language models and then fine-tuned with labeled data. We apply this method to challenging benchmarks in machine translation and abstractive summarization and find that it significantly improves the subsequent supervised models. Our main result is that pretraining improves the generalization of seq2seq models. We achieve state-of-the-art results on the WMT English→German task, surpassing a range of methods using both phrase-based machine translation and neural machine translation. Our method achieves a significant improvement of 1.3 BLEU from the previous best models on both WMT'14 and WMT'15 English→German. We also conduct human evaluations on abstractive summarization and find that our method outperforms a purely supervised learning baseline in a statistically significant manner.

1 Introduction

Sequence to sequence (*seq2seq*) models (Sutskever et al., 2014; Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Allen, 1987; Ñeco and Forcada, 1997) are extremely effective on a variety of tasks that require a mapping between a variable-length input sequence to a variable-length output sequence. The main weakness of sequence to sequence models, and deep networks in general, lies in the fact that they can easily overfit when the amount of supervised training data is small.

In this work, we propose a simple and effective technique for using unsupervised pretraining to improve seq2seq models. Our proposal is to initialize both encoder and decoder networks with pretrained weights of two language models. These pretrained weights are then fine-tuned with the labeled corpus. During the fine-tuning phase, we jointly train the seq2seq objective with the language modeling objectives to prevent overfitting.

We benchmark this method on machine translation for English→German and abstractive summarization on CNN and Daily Mail articles. Our main result is that a seq2seq model, with pretraining, exceeds the strongest possible baseline in both neural machine translation and phrase-based machine translation. Our model obtains an improvement of 1.3 BLEU from the previous best models on both WMT'14 and WMT'15 English→German. On human evaluations for abstractive summarization, we find that our model outperforms a purely supervised baseline, both in terms of correctness and in avoiding unwanted repetition.

We also perform ablation studies to understand the behaviors of the pretraining method. Our study confirms that among many other possible choices of using a language model in seq2seq with attention, the above proposal works best. Our study also shows that, for translation, the main gains come from the improved generalization due to the pretrained features. For summarization, pretraining the encoder gives large improvements, suggesting that the gains come from the improved optimization of the encoder that has been unrolled for hundreds of timesteps. On both tasks, our proposed method always improves generalization on the test sets.

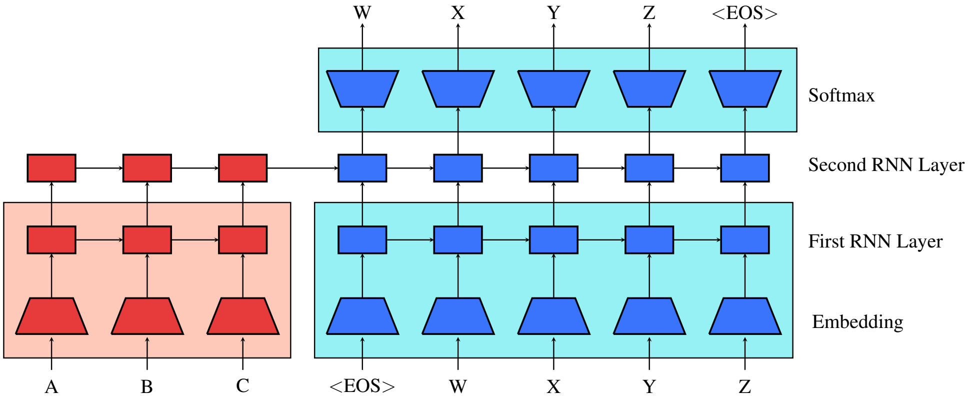


Figure 1: Pretrained sequence to sequence model. The red parameters are the encoder and the blue parameters are the decoder. All parameters in a shaded box are pretrained, either from the source side (light red) or target side (light blue) language model. Otherwise, they are randomly initialized.

2 Methods

In the following section, we will describe our basic unsupervised pretraining procedure for sequence to sequence learning and how to modify sequence to sequence learning to effectively make use of the pretrained weights. We then show several extensions to improve the basic model.

2.1 Basic Procedure

Given an input sequence $x_1, x_2, ..., x_m$ and an output sequence $y_n, y_{n-1}, ..., y_1$, the objective of sequence to sequence learning is to maximize the likelihood $p(y_n, y_{n-1}, ..., y_1|x_1, x_2, ..., x_m)$. Common sequence to sequence learning methods decompose this objective as $p(y_n, y_{n-1}, ..., y_1|x_1, x_2, ..., x_m) = \prod_{t=1}^{n} p(y_t|y_{t-1}, ..., y_1; x_1, x_2, ..., x_m)$.

In sequence to sequence learning, an RNN encoder is used to represent $x_1, ..., x_m$ as a hidden vector, which is given to an RNN decoder to produce the output sequence. Our method is based on the observation that without the encoder, the decoder essentially acts like a language model on y's. Similarly, the encoder with an additional output layer also acts like a language model. Thus it is natural to use trained languages models to initialize the encoder and decoder.

Therefore, the basic procedure of our approach is to pretrain both the seq2seq encoder and decoder networks with language models, which can be trained on large amounts of unlabeled text data. This can be seen in Figure 1, where the parameters in the shaded boxes are pretrained. In the following we will describe the method in detail using machine translation as an example application.

First, two monolingual datasets are collected, one for the source side language, and one for the target side language. A language model (*LM*) is trained on each dataset independently, giving an LM trained on the source side corpus and an LM trained on the target side corpus.

After two language models are trained, a multi-layer seq2seq model M is constructed. The embedding and first LSTM layers of the encoder and decoder are initialized with the pretrained weights. To be even more efficient, the softmax of the decoder is initialized with the softmax of the pretrained target side LM.

2.2 Monolingual language modeling losses

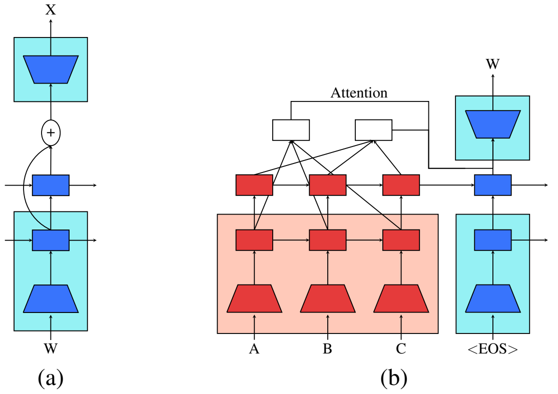
After the seq2seq model M is initialized with the two LMs, it is fine-tuned with a labeled dataset. However, this procedure may lead to *catastrophic forgetting*, where the model's performance on the language modeling tasks falls dramatically after fine-tuning (Goodfellow et al., 2013). This may hamper the model's ability to generalize, especially when trained on small labeled datasets.

To ensure that the model does not overfit the labeled data, we regularize the parameters that were pretrained by continuing to train with the monolingual language modeling losses. The seq2seq and language modeling losses are weighted equally.

In our ablation study, we find that this technique is complementary to pretraining and is important in achieving high performance.

2.3 Other improvements to the model

Pretraining and the monolingual language modeling losses provide the vast majority of improvements to the model. However in early experimentation, we found minor but consistent improvements with two additional techniques: a) residual connections and b) multi-layer attention (see Figure 2).



(a) (b)

Figure 2: Two small improvements to the baseline model: (a) residual connection, and (b) multi-layer attention.

Residual connections: As described, the input vector to the decoder softmax layer is a random vector because the high level (non-first) layers of the LSTM are randomly initialized. This introduces random gradients to the pretrained parameters. To avoid this, we use a residual connection from the output of the first LSTM layer directly to the input of the softmax (see Figure 2-a).

Multi-layer attention: In all our models, we use an attention mechanism (Bahdanau et al., 2015), where the model attends over both top and first layer (see Figure 2-b). More concretely, given a query vector q_t from the decoder, encoder states from the first layer $h_1^1, \ldots, h_T^1$, and encoder states from the last layer $h_1^L, \ldots, h_T^L$, we compute the attention context vector c_t as follows:

$$\alpha_i = \frac{\exp(q_t \cdot h_i^N)}{\sum_{j=1}^{T} \exp(q_t \cdot h_j^N)} \qquad c_t^1 = \sum_{i=1}^{T} \alpha_i h_i^1$$

$$c_t^N = \sum_{i=1}^{T} \alpha_i h_i^N \qquad c_t = [c_t^1; c_t^N]$$

3 Experiments

In the following section, we apply our approach to two important tasks in seq2seq learning: machine translation and abstractive summarization. On each task, we compare against the previous best systems. We also perform ablation experiments to understand the behavior of each component of our method.

3.1 Machine Translation

Dataset and Evaluation: For machine translation, we evaluate our method on the WMT English→German task (Bojar et al., 2015). We used the WMT 14 training dataset, which is slightly smaller than the WMT 15 dataset. Because the dataset has some noisy examples, we used a language detection system to filter the training examples. Sentences pairs where either the source was not English or the target was not German were thrown away. This resulted in around 4 million training examples. Following Sennrich et al. (2015a), we use subword units (Sennrich et al., 2015b) with 89500 merge operations, giving a vocabulary size around 90000. The validation set is the concatenated newstest2012 and newstest2013, and our test sets are newstest2014 and newstest2015. Evaluation on the validation set was with case-sensitive BLEU (Papineni et al., 2002) on tokenized text using `multi-bleu.perl`. Evaluation on the test sets was with case-sensitive BLEU on detokenized text using `mteval-v13a.pl`. The monolingual training datasets are the News Crawl English and German corpora, each of which has more than a billion tokens.

Experimental settings: The language models were trained in the same fashion as (Jozefowicz et al., 2016) We used a 1 layer 4096 dimensional LSTM with the hidden state projected down to 1024 units (Sak et al., 2014) and trained for one week on 32 Tesla K40 GPUs. Our seq2seq model was a 3 layer model, where the second and third layers each have 1000 hidden units. The monolingual objectives, residual connection, and the modified attention were all used. We used the Adam optimizer (Kingma and Ba, 2015) and train with asynchronous SGD on 16 GPUs for speed. We used a learning rate of 5e-5 which is multiplied by 0.8 every 50K steps after an initial 400K steps, gradient clipping with norm 5.0 (Pascanu et al., 2013), and dropout of 0.2 on non-recurrent connections (Zaremba et al., 2014). We used early stopping on validation set perplexity. A beam size of 10 was used for decoding. Our ensemble is con-

		BLEU	
System	*ensemble?*	*newstest2014*	*newstest2015*
Phrase Based MT (Williams et al., 2016)	-	21.9	23.7
Supervised NMT (Jean et al., 2015)	single	-	22.4
Edit Distance Transducer NMT (Stahlberg et al., 2016)	single	21.7	24.1
Edit Distance Transducer NMT (Stahlberg et al., 2016)	ensemble 8	22.9	25.7
Backtranslation (Sennrich et al., 2015a)	single	22.7	25.7
Backtranslation (Sennrich et al., 2015a)	ensemble 4	23.8	26.5
Backtranslation (Sennrich et al., 2015a)	ensemble 12	**24.7**	27.6
No pretraining	single	21.3	24.3
Pretrained seq2seq	single	**24.0**	**27.0**
Pretrained seq2seq	ensemble 5	**24.7**	**28.1**

Table 1: English→German performance on WMT test sets. Our pretrained model outperforms all other models. Note that the model without pretraining uses the LM objective.

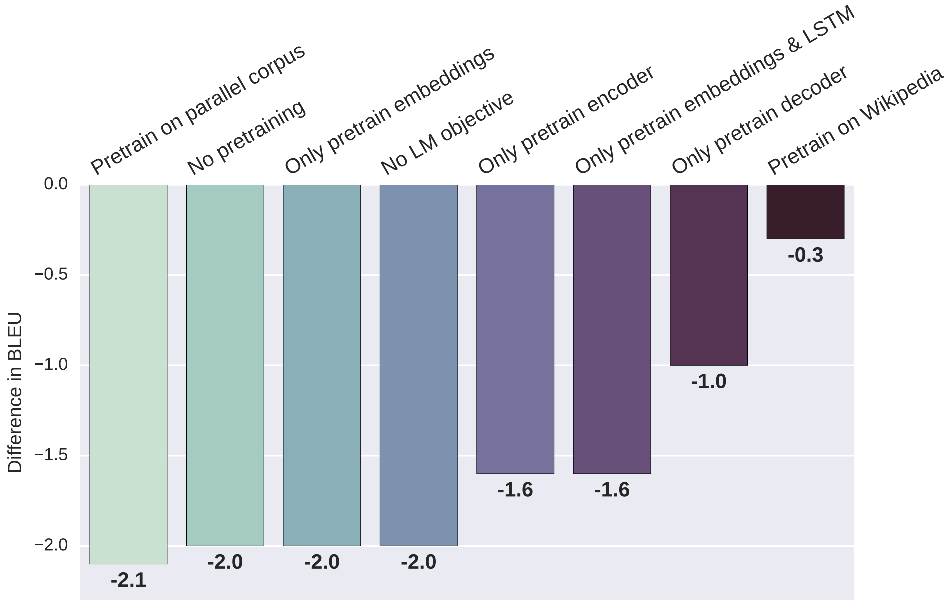


Figure 3: English→German ablation study measuring the difference in validation BLEU between various ablations and the full model. More negative is worse. The full model uses LMs trained with monolingual data to initialize the encoder and decoder, plus the language modeling objective.

structed with the 5 best performing models on the validation set, which are trained with different hyperparameters.

Results: Table 1 shows the results of our method in comparison with other baselines. Our method achieves a new state-of-the-art for single model performance on both newstest2014 and newstest2015, significantly outperforming the competitive semi-supervised *backtranslation* technique (Sennrich et al., 2015a). Equally impressive is the fact that our best single model outperforms the previous state of the art ensemble of 4 models. Our ensemble of 5 models matches or exceeds the previous best ensemble of 12 models.

Ablation study: In order to better understand the effects of pretraining, we conducted an ablation study by modifying the pretraining scheme. We were primarily interested in varying the pretraining scheme and the monolingual language modeling objectives because these two techniques produce the largest gains in the model. Figure 3 shows the drop in validation BLEU of various ablations compared with the full model. The *full model* uses LMs trained with monolingual data to initialize the encoder and decoder, in addition to the language modeling objective. In the following, we interpret the findings of the study. Note that some findings are specific to the translation task.

Given the results from the ablation study, we can make the following observations:

- Only pretraining the decoder is better than only pretraining the encoder: Only pretraining the encoder leads to a 1.6 BLEU point drop while only pretraining the decoder leads to a 1.0 BLEU point drop.
- Pretrain as much as possible because the benefits compound: given the drops of no pretraining at all (-2.0) and only pretraining the encoder (-1.6), the additive estimate of the drop of only pretraining the decoder side is $-2.0 - (-1.6) = -0.4$; however the actual drop is -1.0 which is a much larger drop than the additive estimate.
- Pretraining the softmax is important: Pretraining only the embeddings and first LSTM layer gives a large drop of 1.6 BLEU points.
- The language modeling objective is a strong regularizer: The drop in BLEU points of pretraining the entire model and not using the LM objective is as bad as using the LM objective without pretraining.
- Pretraining on a lot of unlabeled data is essential for learning to extract powerful features: If the model is initialized with LMs that are pretrained on the source part and target part of the *parallel* corpus, the drop in performance is as large as not pretraining at all. However, performance remains strong when pretrained on the large, non-news Wikipedia corpus.

To understand the contributions of unsupervised pretraining vs. supervised training, we track the performance of pretraining as a function of dataset size. For this, we trained a a model with and without pretraining on random subsets of the English→German corpus. Both models use the additional LM objective. The results are summarized in Figure 4. When a 100% of the labeled data is used, the gap between the pretrained and no pretrain model is 2.0 BLEU points. However, that gap grows when less data is available. When trained on 20% of the labeled data, the gap becomes 3.8 BLEU points. This demonstrates that the pretrained models degrade less as the labeled dataset becomes smaller.

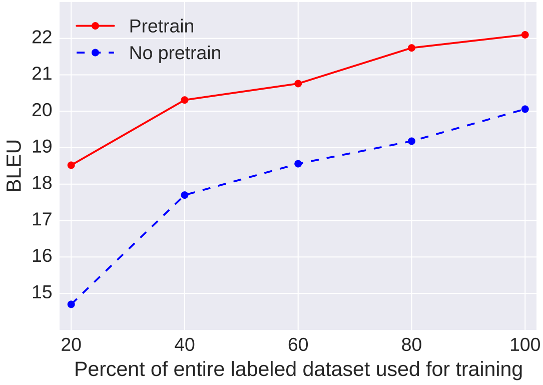


Figure 4: Validation performance of pretraining vs. no pretraining when trained on a subset of the entire labeled dataset for English→German translation.

3.2 Abstractive Summarization

Dataset and Evaluation: For a low-resource abstractive summarization task, we use the CNN/Daily Mail corpus from (Hermann et al., 2015). Following Nallapati et al. (2016), we modify the data collection scripts to restore the bullet point summaries. The task is to predict the bullet point summaries from a news article. The dataset has fewer than 300K document-summary pairs. To compare against Nallapati et al. (2016), we used the anonymized corpus. However, for our ablation study, we used the non-anonymized corpus.[1] We evaluate our system using full length ROUGE (Lin, 2004). For the anonymized corpus in particular, we considered each highlight as a separate sentence following Nallapati et al. (2016). In this setting, we used the English Gigaword corpus (Napoles et al., 2012) as our larger, unlabeled "monolingual" corpus, although all data used in this task is in English.

Experimental settings: We use subword units (Sennrich et al., 2015b) with 31500 merges, resulting in a vocabulary size of about 32000. We use up to the first 600 tokens of the document and

[1] We encourage future researchers to use the non-anonymized version because it is a more realistic summarization setting with a larger vocabulary. Our numbers on the non-anonymized test set are 35.56 ROUGE-1, 14.60 ROUGE-2, and 25.08 ROUGE-L. We did not consider highlights as separate sentences.

System	*ROUGE-1*	*ROUGE-2*	*ROUGE-L*
Seq2seq + pretrained embeddings (Nallapati et al., 2016)	32.49	11.84	29.47
+ temporal attention (Nallapati et al., 2016)	**35.46**	**13.30**	**32.65**
Pretrained seq2seq	32.56	11.89	29.44

Table 2: Results on the anonymized CNN/Daily Mail dataset.

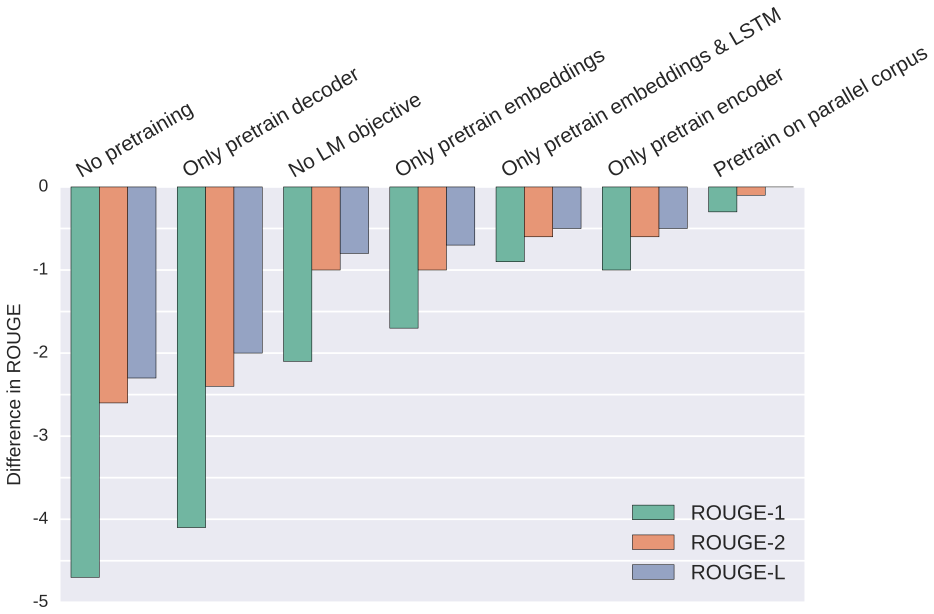


Figure 5: Summarization ablation study measuring the difference in validation ROUGE between various ablations and the full model. More negative is worse. The full model uses LMs trained with unlabeled data to initialize the encoder and decoder, plus the language modeling objective.

predict the entire summary. Only one language model is trained and it is used to initialize both the encoder and decoder, since the source and target languages are the same. However, the encoder and decoder are not tied. The LM is a one-layer LSTM of size 1024 trained in a similar fashion to Jozefowicz et al. (2016). For the seq2seq model, we use the same settings as the machine translation experiments. The only differences are that we use a 2 layer model with the second layer having 1024 hidden units, and that the learning rate is multiplied by 0.8 every 30K steps after an initial 100K steps.

Results: Table 2 summarizes our results on the anonymized version of the corpus. Our pretrained model is only able to match the previous baseline seq2seq of Nallapati et al. (2016). Interestingly, they use pretrained word2vec (Mikolov et al., 2013) vectors to initialize their word embeddings. As we show in our ablation study, just pretraining the embeddings itself gives a large improvement. Furthermore, our model is a unidirectional LSTM while they use a bidirectional LSTM. They also use a longer context of 800 tokens, whereas we used a context of 600 tokens due to GPU memory issues.

Ablation study: We performed an ablation study similar to the one performed on the machine translation model. The results are reported in Figure 5. Here we report the drops on ROUGE-1, ROUGE-2, and ROUGE-L on the non-anonymized validation set.

Given the results from our ablation study, we can make the following observations:

- Pretraining appears to improve optimization: in contrast with the machine translation model, it is more beneficial to only pretrain the encoder than only the decoder of the summarization model. One interpretation is that pretraining enables the gradient to flow much further back in time than randomly initialized weights. This may also explain why pretraining on the parallel corpus is no worse than pretraining on a larger monolingual corpus.

- The language modeling objective is a strong regularizer: A model without the LM objective has a significant drop in ROUGE scores.

Human evaluation: As ROUGE may not be able to capture the quality of summarization, we also performed a small qualitative study to understand the human impression of the summaries produced by different models. We took 200 random documents and compared the performance of a pretrained and non-pretrained system. The document, gold summary, and the two system outputs were presented to a human evaluator who was asked to rate each system output on a scale of 1-5 with 5 being the best score. The system outputs were presented in random order and the evaluator did not know the identity of either output. The evaluator noted if there were repetitive phrases or sentences in either system outputs. Unwanted repetition was also noticed by Nallapati et al. (2016).

Table 3 and 4 show the results of the study. In both cases, the pretrained system outperforms the system without pretraining in a statistically significant manner. The better optimization enabled by pretraining improves the generated summaries and decreases unwanted repetition in the output.

$NP > P$	$NP = P$	$NP < P$
29	88	83

Table 3: The count of how often the no pretrain system (*NP*) achieves a higher, equal, and lower score than the pretrained system (*P*) in the side-by-side study where the human evaluator gave each system a score from 1-5. The sign statistical test gives a p-value of < 0.0001 for rejecting the null hypothesis that there is no difference in the score obtained by either system.

4 Related Work

Unsupervised pretraining has been intensively studied in the past years, most notably is the work by Dahl et al. (2012) who found that pretraining with deep belief networks improved feedforward acoustic models. More recent acoustic models have found pretraining unnecessary (Xiong et al., 2016; Zhang et al., 2016; Chan et al., 2015), probably because the reconstruction objective of deep belief networks is too easy. In contrast, we find that pretraining language models by next step prediction significantly improves seq2seq on challenging real world datasets.

		No pretrain	
		No repeats	Repeats
Pretrain	No repeats	67	65
	Repeats	24	44

Table 4: The count of how often the pretrain and no pretrain systems contain repeated phrases or sentences in their outputs in the side-by-side study. McNemar's test gives a p-value of < 0.0001 for rejecting the null hypothesis that the two systems repeat the same proportion of times. The pretrained system clearly repeats less than the system without pretraining.

Despite its appeal, unsupervised learning has not been widely used to improve supervised training. Dai and Le (2015); Radford et al. (2017) are amongst the rare studies which showed the benefits of pretraining in a semi-supervised learning setting. Their methods are similar to ours except that they did not have a decoder network and thus could not apply to seq2seq learning. Similarly, Zhang and Zong (2016) found it useful to add an additional task of sentence reordering of source-side monolingual data for neural machine translation. Various forms of transfer or multitask learning with seq2seq framework also have the flavors of our algorithm (Zoph et al., 2016; Luong et al., 2015; Firat et al., 2016).

Perhaps most closely related to our method is the work by Gulcehre et al. (2015), who combined a language model with an already trained seq2seq model by fine-tuning additional deep output layers. Empirically, their method produces small improvements over the supervised baseline. We suspect that their method does not produce significant gains because (i) the models are trained independently of each other and are not fine-tuned (ii) the LM is combined with the seq2seq model after the last layer, wasting the benefit of the low level LM features, and (iii) only using the LM on the decoder side. Venugopalan et al. (2016) addressed (i) but still experienced minor improvements. Using pretrained GloVe embedding vectors (Pennington et al., 2014) had more impact.

Related to our approach in principle is the work by Chen et al. (2016) who proposed a two-term, theoretically motivated unsupervised objective for unpaired input-output samples. Though they did not apply their method to seq2seq learning, their framework can be modified to do so. In that case, the first term pushes the output to be highly probable under some scoring model, and the second term ensures that the output depends on the input. In the seq2seq setting, we interpret the first term as a pretrained language model scoring the output sequence. In our work, we fold the pretrained language model into the decoder. We believe that using the pretrained language model only for scoring is less efficient that using all the pretrained weights. Our use of labeled examples satisfies the second term. These connections provide a theoretical grounding for our work.

In our experiments, we benchmark our method on machine translation, where other unsupervised methods are shown to give promising results (Sennrich et al., 2015a; Cheng et al., 2016). In back-translation (Sennrich et al., 2015a), the trained model is used to decode unlabeled data to yield extra labeled data. One can argue that this method may not have a natural analogue to other tasks such as summarization. We note that their technique is complementary to ours, and may lead to additional gains in machine translation. The method of using autoencoders in Cheng et al. (2016) is promising, though it can be argued that autoencoding is an easy objective and language modeling may force the unsupervised models to learn better features.

5 Conclusion

We presented a novel unsupervised pretraining method to improve sequence to sequence learning. The method can aid in both generalization and optimization. Our scheme involves pretraining two language models in the source and target domain, and initializing the embeddings, first LSTM layers, and softmax of a sequence to sequence model with the weights of the language models. Using our method, we achieved state-of-the-art machine translation results on both WMT'14 and WMT'15 English to German. A key advantage of this technique is that it is flexible and can be applied to a large variety of tasks.

References

Robert B. Allen. 1987. Several studies on natural language and back-propagation. *IEEE First International Conference on Neural Networks*.

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

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*.

Jianshu Chen, Po-Sen Huang, Xiaodong He, Jianfeng Gao, and Li Deng. 2016. Unsupervised learning of predictors from unpaired input-output samples. abs/1606.04646.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

G. E. Dahl, D. Yu, L. Deng, and A. Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *NIPS*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman-Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multilingual neural machine translation. *arXiv preprint arXiv:1606.04164*.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence RNNs for text summarization. *arXiv preprint arXiv:1602.06023*.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. ACL.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *ICML*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Hasim Sak, Andrew W. Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Felix Stahlberg, Eva Hasler, and Bill Byrne. 2016. The edit distance transducer in action: The university of cambridge english-german system at wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 377–384, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. 2016. Improving LSTM-based video description with linguistic knowledge mined from text. *arXiv preprint arXiv:1604.01729*.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. 2016. Edinburgh's statistical machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 399–410, Berlin, Germany. Association for Computational Linguistics.

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2016. Achieving human parity in conversational speech recognition. abs/1610.05256.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*.

Yu Zhang, William Chan, and Navdeep Jaitly. 2016. Very deep convolutional networks for end-to-end speech recognition. abs/1610.03022.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *EMNLP*.

Ramon P. Ñeco and Mikel L. Forcada. 1997. Asynchronous translations with recurrent neural nets. *Neural Networks*.