Modified Sequence to Sequence model using Recurrent Memory Network

Sang Jun Yum
New York University
sjy269@nyu.edu

Yurui Mu
New York University
ym1495@nyu.edu

Abstract

In this paper, we propose a modified machine translation model based on RNN Encoder-Decoder (Cho et al., 2014) and Recurrent Memory Network (Tran et al., 2016). The original Encoder-Decoder is consisted with two recurrent networks (RNN); one encodes a sentence written in the source language into a fixed sized context vector, and the other takes symbols written in target language along with the context vector from the encoder to generate sequence of symbols in target language. These two networks are jointly trained to maximize the probability of target sequence conditioned on source sequence. In our experiment, we replaced the decoder of the original Encoder-Decoder with Recurrent Memory Network (Tran et al., 2016), and observed the performance of the model by comparing it to the original Encoder-Decoder.

1 Introduction

It has been proven that RNN (Mikolov et al., 2010) is a powerful tool for various Natural Language Processing tasks. It is especially the case when it comes to language modeling and machine translation. Through RNN, information from previous symbols are passed to the current symbol, enabling the language model to compute the probability distribution of the next symbol conditioned on symbols that precede the current symbol. As briefly explained, encoder is a RNN that takes the source sentence and maps it to a fixed-length context vector. Starting from start of sentence token, each symbol in the target sentence is fed into the decoder along with the context vector from encoder to compute the probability distribution of a next target word conditioned on previous target words and source sentence. In the perspective of language modeling, except that it takes the encoder context into the consideration, decoder is a species of a RNN language model. (Cho et al., 2014) observed that using Gated Recurrent Unit instead of non-gated RNN is desirable in language modeling and machine translation task. The idea of gating connects to LSTM. (Bowman et al., 2015) and (Filippova et al., 2015) hypothesized that LSTM is a powerful tool when it comes to capturing syntactic aspects of sentences. We expected replacing the decoder with LSTM would increase the performance of translation via increasing the models capacity to learn the syntactic structure of target sentences. To let the translation model to learn the underlying dependencies in target sentences, the LSTM Decoder is enhanced using Recurrent Memory Network (Tran et al., 2016). Recurrent Memory Network is a variation of a LSTM language model that attends to recent few words. Thus, in our proposed model, the decoder not only takes previous target words and source sentence into the consideration, but also the attention over history of recent few target words. The following is the list of experiments and implementations made in this paper.

1. We implemented a simple Encoder-Decoder and observed the performance on a relatively big data set with complex and long sentences (IWSLT 2016 English to German), and a relatively small data set with short and simple sentences (Multi30k English to German).

2. We implemented an Encoder-Decoder of which Decoder is replaced with RMN to observe the performance on the same data sets (IWSLT and Multi30k).

2 Preliminaries

2.1 Recurrent Neural Network

A Recurrent neural network (RNN) is a neural network that is consisted of hidden state **h** and a variable length sequence **x** = $(x_1,...,x_T)$.

$$h_t = g(U * h_{t-1} + W * x_t + b) \quad (1)$$

where g is a non linearity. From the equation above, it is observable that hidden state of each time step of RNN is affected by the previous time steps. Thus, the output at each time step t is the conditional distribution $p(x_t|x_{<t})$, which is by Bayes' Rule,

$$p(X) = \prod_{t=1}^{T} p(x_t|x_{<t}) \quad (2)$$

And each term in the equation above can be computed by softmax activation function.

$$p(x_{t,j}) = \frac{\exp(w_j h_t)}{\sum_{j'=1}^{K} \exp(w_{j'} h_t)} \quad (3)$$

2.2 Gated Recurrent Unit and Long Short Memory Network

Instead of using a regular, non-gated RNN, using Gated Recurrent Unit (GRU) is crucial in translation task. GRU can effectively allow the hidden state to retain the information from the previous hidden state and newly created hidden state. Update gate decides whether the hidden state is to be updated with newly created hidden state, $\tilde{h}$. Reset gate decides whether the previous hidden state is to be used in computing the current hidden state.

$$u_t = sigmoid(W_u x_t + U_u h_t) \quad (4)$$

$$r_t = sigmoid(W_r x_t + U_r h_t) \quad (5)$$

$$\tilde{h_t} = tanh(W x_t + U(r_t \odot h_t)) \quad (6)$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot \tilde{h_t} \quad (7)$$

In Long Short Term Memory, the cell state of the network is explicitly hidden so that only output state(h) of the network is visible to other part of the network.

$$i_t = sigmoid(W_i x_t + U_i h_t) \quad (8)$$

$$f_t = sigmoid(W_f x_t + U_f h_t) \quad (9)$$

$$o_t = sigmoid(W_o x_t + U_o h_t) \quad (10)$$

$$c_t = c_{t-1} \odot f_t + i_t \odot \tilde{c_t} \quad (11)$$

$$h_t = tanh(c_t) \odot o_t \quad (12)$$

LSTM is especially powerful tool for capturing the long-term dependencies among the memories.

2.3 Encoder Decoder

Our baseline model (Encoder Decoder) is based on the model introduced (Cho et al., 2014). The model is comprised of two seperate RNNs. The first one, encoder, takes the variable length source sentence to output a fixed size representation of the source sentence, c. This vector is often referred as a *context vector* of the source sentence.

$$c = h_t = g(U * h_{t-1} + W * x_t + b_{enc}) \quad (13)$$

Decoder is another RNN that takes target word one by one to output a prediction for next word for each source word; as explained earlier in introduction, such behavior is equivalent to the typical RNN language modeling task. Nonetheless, as we desire the probability of next target word to be conditioned on not only previous target words, but also the information from the source sentence, each target word, before it is fed into the decoder, is concatenated with the encoder context c.

$$z_t = g(U * z_{t-1} + W * [y_t; c] + b_{dec}) \quad (14)$$

The conditional probability of the next symbol can then be simply expressed as

$$P(y_t|y_{t-1}, y_{t-2}, ..., y_1, X) = g(z_t, y_{t-1}, X) \quad (15)$$

where X is the source sentence.

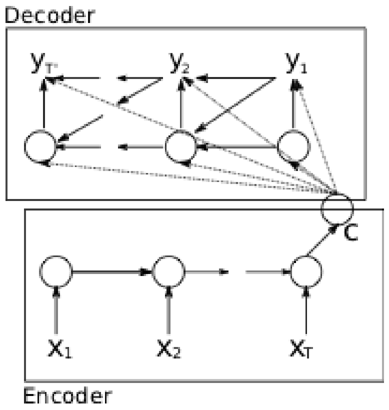

Figure 1: RNN Encoder Decoder(Cho et al., 2014)

2.4 Recurrent Memory Network

In this section, we introduce Recurrent Memory Network (RMN), a new LSTM language model that replaces the decoder of our base line sequence to sequence model. The motivation of RMN is to facilitate analyzing what information is exactly contained in decoder's hidden states. RMN is expected to capture which information is contained in the hidden states, as well as discovering the dependencies among them.

RMN consists of two components: a LSTM and a Memory Block (MB). The LSTM simply replace the RNN component of the base line decoder. It takes a target word concatenated with the encoder context as an input. The MB takes the hidden state of the LSTM and compares it to the few most recent inputs (memory) using an attention mechanism (Bahdanau et al., 2014). By doing so, a trained model can give us insight into the information that is retained over time in the LSTM.

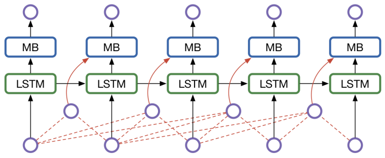

Figure 2: Recurrent Memory Network(Tran et al., 2016)

2.4.1 Memory Block

At time step *t*, memory block takes two inputs: LSTM's hidden state z_t and a set of *n* most recent words, $\{y_i\}$. In this paper, $\{y_i\}$ is referred as memory, and *n* is referred as the memory size. In MB, there are two embedding matrices, M and C, both of which dimension is $|V|$ by d where $|V|$ is the size of the target language vocabulary and d is the embedding dimension of every word. The memory is embedded twice using each of these embedding matrices. The embedded representations of memory are written as $M_i = M(\{y_i\})$ and $C_i = C(\{y_i\})$. M_i is used to compute an attention distribution over the memory.

$$P_t = softmax(M_i z_t) \tag{16}$$

The context vector representation of memory is then computed as following.

$$s_t = C_i^T P_t \tag{17}$$

The context vector representation of memory and the LSTM hidden state z_t are combined by gated recurrent function $g(.)$ (explained in 2.2) to obtain the output of the memory block.

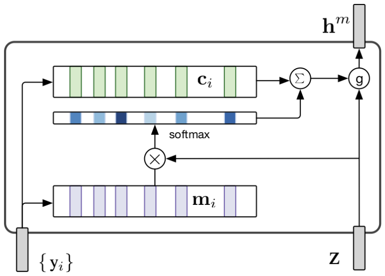

Figure 3: Recurrent Memory Network(Tran et al., 2016)

The output of the memory block is then concatenated with the encoder context vector c, then is affine-transformed to be input into another softmax to output the conditional probability of next target word, $p(y_{t+1}|y_{t-n}, y_{t-n+1}, ..., y_t, \{y_i\}, X)$. where X is the source sentence.

3 Empirical Study

3.1 Data set

In our experiment, we trained our models with two different data sets:

1. IWSLT (The International Workshop on Spoken Language Translation) 2016 TED Talk English-German Machine Translation Dataset.

2. Multi30k: Multilingual English-German Image Descriptions.(Elliott et al., 2016)

We processed texts using SpaCy package, which gives the fastest syntactic parser in the world and top

1% highest accuracy.(Choi et al., 2015) The table 1 shows the sizes of two dataset after using SpaCy tokenizers. To include words that have appeared at least twice, we set the maximum vocabulary size for each dataset: 50K and 10K.

Dataset	Train	Dev	Test	$\lVert V_{en}\rVert$	$\lVert V_{de}\rVert$
IWSLT	196884	993	1305	58k	125k
Multi30k	29000	1014	1000	10k	18k

Table 1: Dataset

3.2 Training and Inference

Training is done using Ada delta optimizer; we minimized the negative log likelihood between the translated word and ground-truth probability of next word. Due to the limited computational resources, we took greedy method (Beam search with $k = 1$) (Wiseman and Rush, 2016) for the inference.

3.3 Evaluation

Among several Machine Translation evaluation metrics, we used BLEU (Papineni et al., 2002) as our evaluation metric. BLEU (Bilingual Evaluation Understudy) is one of the first metrics to claim a high correlation with human judgements of quality, and remains one of the most popular automated and inexpensive metrics(Ke and Ma, 2014). BLEU is computed as following.

$$BLEU = BP \cdot exp(\sum_{i=1}^{n} w_n log(p_n))$$

where

$$BP = \begin{cases} 1 & c > r \\ e^{1-r/c} & c \leq r \end{cases} \tag{18}$$

In our report, we report the document-level BLEU from Moses (Koehn et al., 2007) and NLTK average sentence-level BLEU with smoothing technique from (Chen and Cherry, 2014). This smoothing technique prevents precision of shorter translation from being inflated.

3.4 Hyper Parameter Tuning and Early stopping

For our implementation, hyper-parameters (learning rate, word embedding dimension and RNN hidden dimension) are tuned using Random Search. (Bergstra and Bengio, 2012) Parameter sets are selected randomly from uniformly constructed parameter combinations. We implemented early stopping using validation BLEU score (from Moses script) as the validation metric.

3.5 Experiment Setup

In our implementation of encoder, we used 2-layered uni-directional gated recurrent unit to encode the source sentence. The decoder is implemented using single layered uni-directional GRU. In our implementation of recurrent memory network (RMN), GRU is replaced with LSTM. At each time step, attention over 7 previous target words (memory) is combined with hidden state of LSTM. This output is again concatenated with encoder context, then affine-transformed to be inputted to a softmax to output conditional distribution of next target word.

$$p(y_t|y_1, y_2, ..., y_{t-1}, \{y_i\}, X) \tag{19}$$

4 Experiment Results and Analysis

Table 2 and table 3 show the experiment results from Multi30k data set and IWSLT data set. For our Multi30k task, we used 10k vocabulary for both English and German, taking all the words which have appeared at least twice into consideration. To increase training speed, we used 10 as batch size. Both embedding dimension and hidden size were set to 1024.

As for our IWSLT task, we used 58k vocabulary for both English and German, including all the words occurring at least twice in English and German. We set 12 as batch size, 2 hidden layers in Encoder GRU, 128 as embedding size and 1000 as hidden size in both RNN and RMN decoder, with memory size 7 in RMN memory block.

Model	$BLEU_{NLTK}$	$BLEU_{Moses}$
Encoder-Decoder	33.71	29.01
RMN Decoder	30.96	24.24

Table 2: Test BLEUs on Multi30k Dataset

4.1 Experiment Analysis

4.1.1 Multi30k Experiment

We could witness that in experiments using Multi30k, our model performances nearly reached

Model	$BLEU_{NLTK}$	$BLEU_{Moses}$
Encoder-Decoder	22.93	10.15
RMN Decoder	23.64	9.13

Table 3: Test BLEUs on IWSLT Dataset

the the performance from the original experiment (Cho et al., 2014). One notable thing in Multi30k experiments is that our base line model (encoder decoder) outperformed our modified model using RMN. The decreased performance on the modified model is likely due to information overflow and contamination. LSTM of RMN takes target word that is concatenated with encoder context. The output (hidden) of LSTM is fed into the memory block. Before the final softmax layer, memory block output that conveys information regarding target word history (memory) and current target word (that also conveys information about the source sentence) is again concatenated with the encoder context. It is likely that such process of multiple concatenation of encoder context vector contaminated the representation of memory block output, resulting in degenerated performance.

Another possible reason of information contamination is that the information regarding memory (target word history) is not a highly relevant or useful information for translation. It is possible that adding this information resulted in giving unnecessary information to the model, resulting in information contamination. In short, there are two possible reasons in degeneration.

1. Concatenating our vector representation from the modified decoder with encoder context multiple times contaminated relevent information.

2. Information about previous target word history has very little relevancy to the translation task.

4.1.2 IWSLT Experiment

In our experiment using IWSLT data set, we could not produce satisfactory test results that are close to results from original implementation. Notice that compared to dictionary size (15K) used for experiment performed in (Cho et al., 2014), IWSLT data set houses more than 50K (English) and 120K (German) vocabularies in the dictionary. In order to obtain word vector representations that successfully embed words from vocabularies of such a huge dictionary sizes, the embedding dimension has to grow over 500 and 1000. Even if the maximum length of sentence is limited, such embedding dimension resulted in GPU memory overflow. In order to perform an adequate experiment, we need to process the IWSLT data set into more reasonable size such that reasonably sized vocbabulary spans over 90% of the whole data set.

4.2 Qualitative Analysis

From 4, we could observe that both models perform well for short sentences. Nonetheless, for long sentences, the number of overlapping n-gram (especially with high n) between hypothesis and reference is observed to decrease even though there are still some unigrams that overlap. To increase the number of overlapping n-gram with high n, it is highly recommended to use beam search, instead of taking a greedy method.

5 Conclusions and Future Improvement

5.1 Conclusion

We implemented a base line sequence to sequence model and a modified machine translation model. The modified model failed to witness any performance improvements over the base line, possibly due to reasons stated in 4.1.1. There are, however, various possible ways of improving both base-line and modified model to maximize performances.

5.2 Future Improvement

The followings are possible future improvement of our model.

1. We could use bi-directional encoder, and use soft attention.

2. (Wu et al., 2016) implemented a sequence to sequence model using deep LSTM with multiple layers. If enough computational resource is accessible, we could use deeper GRU and LSTM for our implementation.

3. Instead of taking greedy method, we could use beam search(Wiseman and Rush, 2016) to select top K combinations of words with highest probability for each word translation. This

	Texts
English Source	Man running wearing a blue shirt with a number taped to it .
German Reference	Ein Lufer in einem blauen Shirt mit einer ¡unk¿ Nummer .
RNN Hypothesis	Ein Mann mit einem blauen Oberteil und mit einem blauen Bandana -
RMN Hypothesis	Der Mann in einem blauen Oberteil und mit einer hellblauen Mtze trgt
English Source	People sitting in a circle outside a large building .
German Reference	Menschen , die vor einem groen Gebude im Kreis sitzen .
RNN Hypothesis	Menschen sitzen im Freien vor einem groen Gebude .
RMN Hypothesis	Mehrere Personen sitzen drauen in einem groen Gebude .
English Source	An African American man walking down the street .
German Reference	Ein Afroamerikaner geht die Strae hinunter .
RNN Hypothesis	Ein Afroamerikaner geht die Strae entlang .
RMN Hypothesis	Ein Afroamerikaner geht die Strae entlang .
English Source	A man in a black jacket and checkered hat wearing black and white striped pants plays an electric guitar on a stage with a singer and another guitar player in the background .
German Reference	Ein Mann mit kariertem Hut in einer schwarzen Jacke und einer schwarz - wei gestreiften Hose spielt auf einer Bhne mit einem Snger und einem weiteren Gitarristen im Hintergrund auf einer E - Gitarre .
RNN Hypothesis	Ein Mann in Schwarz und schwarzen Hosen spielt auf einer elektrischen Gitarre und spielt mit einem anderen Typen im Hintergrund .
RMN Hypothesis	Ein Mann in einer schwarz und einem schwarzen Jacke und einem schwarzen Weste spielt auf einem hellbraunen Gitarre und eine Gitarre Gitarre , Gitarre auf den schwarzen ¡unk¿ spielt .

Table 4: Random Translation Results on Multi30k Dataset

is especially the case for translation of longer sentences.

4. It is possible to implement RMN such that we only concatenate the encoder context once (either with target word vector before LSTM or with MB output before softmax) to prevent information contamination.

5. (Tran et al., 2016) implemented Recurrent Memory Recurrent, which is an architecture that contains another LSTM on top of Memory Block to enable interaction between memory blocks. In our future experiments, we could add this additional LSTM in our model.

6 Contribution and Work Distribution

1. Sang Jun Yum: model implementation, data preparation, debugging, reprot writing, training/testing/validation loop implementation.

2. Yurui Mu: proposal writing, training loop implementation, hyper-parameter turning and early stopping implementation, code deployment, record keeping.

Code to our project can be found at `https://github.com/ysangj/NLP_DeepLearning_final_project`

Acknowledgments

We appreciate help and guidance from Professor Sam Bowman, Professor Kyunghyun Cho, Artie Shen, Kelly Zhang and Nikita Nangia.

References

[Bergstra and Bengio2012] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, February.

[Bowman et al.2015] Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. In *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches - Volume 1583*, COCO'15, pages 37–42, Aachen, Germany, Germany. CEUR-WS.org.

[Chen and Cherry2014] Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *WMT@ACL*.

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

[Choi et al.2015] Jinho D. Choi, Joel R. Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *ACL*.

[Elliott et al.2016] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *CoRR*, abs/1605.00459.

[Filippova et al.2015] Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*.

[Ke and Ma2014] Xiaohua Ke and Qinghua Ma. 2014. Study on an impersonal evaluation system for english-chinese translation based on semantic understanding. *Perspectives*, 22(2):242–254.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mikolov et al.2010] Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. 2:1045–1048, 01.

[Papineni et al.2002] 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 on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Tran et al.2016] Ke M. Tran, Arianna Bisazza, and Christof Monz. 2016. Recurrent memory network for language modeling. *CoRR*, abs/1601.01272.

[Wiseman and Rush2016] Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *CoRR*, abs/1606.02960.

[Wu et al.2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.