Global-Context Neural Machine Translation through Target-Side Attentive Residual Connections

Lesly Miculicich Werlen 1,2, Nikolaos Pappas 1, Dhananjay Ram 1,2, Andrei Popescu-Belis 3
1Idiap Research Institute, Martigny, Switzerland, 2École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland, 3HEIG-VD / HES-SO, Yverdon, Switzerland
{lmiculicich, npappas, dram}@idiap.ch, andrei.popescu-belis@heig-vd.ch

Abstract

Neural sequence-to-sequence models achieve remarkable performance not only in machine translation (MT) but also in other language processing tasks. One of the reasons for their effectiveness is the ability of their decoder to capture contextual information through its recurrent layer. However, its sequential modeling over-emphasizes the local context, i.e. the previously translated word in the case of MT. As a result, the model ignores important information from the global context of the translation. In this paper, we address this limitation by introducing attentive residual connections from the previously translated words to the output of the decoder, which enable the learning of longer-range dependencies between words. The proposed model can emphasize any of the previously translated words, as opposed to only the last one, gaining access to the global context of the translated text. The model outperforms strong neural MT baselines on three language pairs, as well as a neural language modeling baseline. The analysis of the attention learned by the decoder confirms that it emphasizes a wide context, and reveals resemblance to syntactic-like structures.

Introduction

Neural machine translation (Kalchbrenner and Blunsom 2013; Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Bahdanau, Cho, and Bengio 2015) has recently become the state-of-the-art approach to machine translation (Bojar and others 2016). Due to its quality and simplicity, in contrast to the previous phrase-based statistical machine translation approaches (Koehn 2010), neural machine translation (NMT) has been adopted widely in both the academia and industry. The effectiveness of the attention-based NMT can be largely attributed to the ability of its decoder to capture contextual information through its recurrent layer. However, its sequential modeling over-emphasizes the local context, mainly the previously translated word, and, consequently, gives insufficient importance to the global context of a translated text. Ignoring the global context may lead to sub-optimal lexical choices and imperfect fluency of an output text.

To address this limitation, we propose a novel approach, represented in Figure 1b which is based on attentive residual connections from the previously translated words to the output of the decoder, within an attention-based NMT architecture. The ability of the model to access to the global context at any time improves the flow of information and enables the learning of longer-range dependencies between words than previous models. Moreover, these benefits require only a small computational overhead with respect to previous models. We demonstrate the generality of the proposed approach by applying it, in addition to NMT, to language modeling. For both tasks, our approach outperforms strong baselines, respectively in terms of BLEU (Papineni et al. 2002) and perplexity scores.

Figure 1: Comparison between the decoder of the baseline NMT and the proposed decoder with extended context.
Our contributions are the following ones:

- We propose and compare several options for using attentional residual connections within a standard decoder for attention-based NMT, which enable access to the global context of a translated text.
- We demonstrate consistent improvements over strong SMT and NMT baselines for three language pairs (English-to-Chinese, Spanish-to-English, and English-to-German), as well as for language modeling.
- We perform an ablation study and analyze the learned attention function in terms of its behavior and structure, providing additional insights on its actual contributions.

**Related Work**

Enhancing sequential models in order to better capture the structure of sentences has been explored in a variety of studies. Many of them investigated recursive long-term memory (LSTM) (Hochreiter and Schmidhuber 1997) architectures to combine words into phrases and higher-level arrangements for generating sentences (Zhu, Sobhansi, and Guo 2015; Le and Zuidema 2015; Chen et al. 2016). More specifically, Tai, Socher, and Manning (2015) proposed a tree-structure LSTM to model syntactic properties of a sentence. Other architectures were designed to model sentences as graphs (Zhu, Sobhansi, and Guo 2016), or used a self-attention memory to capture word dependencies or relations in a sentence (Cheng, Dong, and Lapata 2016). In contrast to the latter study, which modified the LSTM units using memory networks, the self-attention mechanism proposed here uses residual connections and does not interfere with the recurrent layer, thus preserving its efficiency. For instance, in the tree-to-sequence approach proposed by Eriguchi, Hashimoto, and Tsuruoka (2016), the encoder is modified to use information from the syntactic tree of the source sentence. In the string-to-tree model proposed by Aharoni and Goldberg (2017), linearized trees are used to incorporate structural information of the source and target sentences in a sequential model. In the sequence-to-dependency method proposed by Wu et al. (2017), a dependency tree is used on the decoder side. All these approaches utilize external parsers to obtain structural information. In the present work, we let the network learn meaningful connections by itself. The dependencies that appear to be learned are not strictly syntactic as the ones obtained from a parser, but are important for optimizing the objective function of the translation model.

To account for longer-range dependencies, our model relies on residual connections, also called shortcut connections, which have been shown to improve the learning process of deep neural networks by addressing the vanishing gradient problem (He et al. 2016). These connections create a direct path from previous layers, which helps in the transmission of information. Recently, several architectures using residual connections with LSTMs have been proposed (Zhang et al. 2016; Kim, El-Khamy, and Lee 2017), applying the shortcut connections between LSTM layers in the spatial domain. In contrast, here, we propose to use residual connections in the temporal domain.

**Background: Neural Machine Translation**

Neural machine translation aims to compute the conditional distribution of emitting a sentence in a target language given a sentence in a source language, noted \( p_\Theta(y|x) \), where \( \Theta \) is the set of parameters of the neural model, and \( y = \{y_1, \ldots, y_n\} \) and \( x = \{x_1, \ldots, x_m\} \) are respectively the representations of source and target sentences as sequences of words. The parameters \( \Theta \) are learned by training a sequence-to-sequence neural model on a corpus of parallel sentences. In particular, the learning objective is typically to maximize the following conditional log-likelihood:

\[
\max_{\Theta} \frac{1}{N} \sum_{n=1}^{N} \log(p_\Theta(y|x)) \quad (1)
\]

The attention-based NMT model designed by Bahdanau, Cho, and Bengio (2015) has become the de-facto baseline for machine translation. The model is based on RNNs, typically using gated recurrent units (GRUs) as in the system designed by Cho et al. (2014) or LSTMs (Hochreiter and Schmidhuber 1997). The architecture is comprised of three main components: an encoder, a decoder, and an attention mechanism.

The goal of the encoder is to build meaningful representations of the source sentences. It consists of a bidirectional RNN which includes contextual information from past and future words into the vector representation \( h_t \) of a particular word vector \( x_t \), formally defined as follows:

\[
h_t = \overrightarrow{h_t}, \overleftarrow{h_t} \quad (2)
\]

\[
\overrightarrow{h_t} = f(x_t, h_{t-1}) \quad (3)
\]

\[
\overleftarrow{h_t} = f(x_t, h_{t+1}) \quad (4)
\]

Here, \( \overrightarrow{h_t} \) and \( \overleftarrow{h_t} \) are the hidden states of the forward and backward passes of the bidirectional RNN respectively, and \( f \) is a non-linear activation function.

The decoder (Figure 1a) is in essence a recurrent language model. At each time step, it predicts a target word \( y_t \) conditioned over the previous words and the information from the encoder using the following posterior probability:

\[
p(y_t|y_1, \ldots, y_{t-1}, c_t) \approx g(s_t, y_{t-1}, c_t) \quad (5)
\]

where \( g \) is a non-linear multilayer function that outputs the probability of \( y_t \).

The hidden state of the decoder \( s_t \) is defined as:

\[
s_t = f(s_{t-1}, y_{t-1}, c_t) \quad (6)
\]

It depends on a context vector \( c_t \) that is learned by the attention mechanism.

The attention mechanism allows the decoder to select which parts of the source sentence are more useful to predict the next output word. This goal is achieved by considering a weighted sum over all hidden states of the encoder as follows:

\[
c_t = \sum_{i=1}^{m} \alpha_i h_i \quad (7)
\]
where $\alpha_t^i$ is a weight calculated using a normalized exponential function, also known as alignment function, which computes how good is the match between the input at position $i$ and the output at position $t$:

$$\alpha_t^i = \frac{\exp(e_t^i)}{\sum_j \exp(e_j^t)}$$

where

$$e_t^i = a(s_{t-1}, h_i)$$

Different types of alignment functions have been used for the NMT framework, as investigated by [Luong, Pham, and Manning2015]. Here, we use the one originally defined by [Bahdanau, Cho, and Bengio2015].

**Decoder with Attentive Residual Connections**

The state-of-the-art decoder of the attention-based NMT model uses a residual connection from the previously predicted words to the output classifier in order to enhance the performance of translation. As we can see in Equation [5] the probability of prediction of a particular word is calculated by a function $g$ which takes as input the hidden state of the recurrent layer $s_t$, the representation of the previously predicted word $y_{t-1}$, and the context vector $c_t$. In theory, $s_t$ should be enough for predicting the next word given that it is dependent on the other two local-context components according to Equation [6]. However, this model overemphasizes the local context (through $c_t$ and $y_t$), and essentially, the last predicted word plays the most important role for generating the next word. How can we allow the model to exploit a broader context for translation?

To answer this question, we propose to change the decoder formula by including residual connections not only from the previous time step $y_{t-1}$, but from all previous time steps from $y_0$ to $y_{t-1}$. This reinforces the memory of the recurrent layer towards what has been translated so far, through weighted connections applied to all previously predicted words. At each time step, the new model decides which of the previously predicted words should be emphasized to predict the next one. In order to deal with the dynamic length of this new input, we use a target-side summary vector $d_t$ that can be interpreted as the representation of the decoded sentence until the time $t$ in a word embedding space. We therefore modify Equation [5] replacing $y_{t-1}$ with $d_t$:

$$p(y_t|y_1, ..., y_{t-1}, c_t) \approx g(s_t, d_t, c_t)$$

The replacement of $y_{t-1}$ with $d_t$ means that the number of parameters added to the model is dependent only on the calculation of $d_t$. Figure [11] illustrates the modification made to the decoder. We define two methods for summarizing the context into $d_t$, which are described in the following sections.

**Mean Residual Connections**

One simple way to aggregate information from multiple word embeddings is by averaging them. The average of the embedding vectors represents a shared space of the previously decoded words, which can be understood as the sentence representation until time $t$. We hypothesize that this representation is more informative than using only the embedding of the previous word. Formally, this representation is computed as follows:

$$d_t^{avg} = \frac{1}{t-1} \sum_{i=1}^{t-1} y_i$$

**Attentive Residual Connections**

Averaging is a simple and cheap way to aggregate information from multiple words, but may not be sufficient for all dependencies. Instead, we propose a dynamic way to aggregate information in each sentence, such that different words have different importance according to their relation with the prediction of the next word. We propose to use a shared self-attention mechanism to obtain a summary representation of the translation, i.e. a weighted average representation of the words translated from $y_0$ to $y_{t-1}$. This mechanism aims to model, in part, important non-sequential dependencies among words, and serves as a complementary memory to the recurrent layer.

$$d_t^{avg} = \sum_{i=1}^{t-1} \alpha_t^i y_i$$

where

$$\alpha_t^i = \frac{\exp(e_t^i)}{\sum_{j=1}^{t-1} \exp(e_j^t)}$$

The weights of the attention model are computed by a scoring function $e_t^i$ that predicts how important each previous word ($y_0, ..., y_{t-1}$) is for the current prediction $y_t$. Figure [11] illustrates the attentive residual connections in the proposed decoder at the moment of prediction of $y_t$, in contrast with the baseline model presented in Figure [1a]. We experiment with four different scoring functions:

$$e_t^i = \begin{cases} 
v^T \tanh(W_y y_t) & \text{simple} \\
v^T \tanh(W_y y_t + W_c c_t) & \text{sum} \\
v^T \tanh(W_y y_t + W_c c_t) & \text{product} \\
y_t W_c c_t & \text{dot product}
\end{cases}$$

where $v \in \mathbb{R}^e$, $W_y \in \mathbb{R}^{e \times e}$, and $W_c \in \mathbb{R}^{e \times c}$ are parameters of the network; $e$ and $c$ are dimensions of the word embedding, and contextual vector respectively.

Firstly, we explore a simple scoring function that is calculated based only on the previous hidden states of the decoder as used e.g. by [Pappas and Popescu-Belis2017]. Secondly, we study the same scoring function (noted sum) as for the attention mechanism to the encoder, as proposed by [Bahdanau, Cho, and Bengio2015]. Thirdly, we consider an element-wise product, and lastly a dot product function as proposed by [Luong, Pham, and Manning2015]. In contrast to the first attention function which is based only on the previous word, the other three functions make use of the context vector $c_t$. 
Experimental Data and Setup

Datasets

To evaluate the proposed MT models in different conditions, we select three language pairs with increasing amounts of training data: English-Chinese (0.5M sentences), Spanish-English (2.1M), and English-German (4.5M).

For English-to-Chinese, we use a subset of the UN parallel corpus (Rafalovitch and others 2009) with 500,000 sentence pairs for training, 2,000 sentences for development, and 2,000 sentences for testing. For training Spanish-to-English MT, we use a subset of WMT 2013 (Bojar and others 2013) corresponding to Europarl v7 with ca. 1,900,000 sentence pairs, plus News Commentary v11 with ca. 200,000 pairs. Newstest2012 and Newstest2013 were used for development and testing respectively, with 3,000 sentence pairs each. Finally, we use the complete English-to-German set from WMT 2016 (Bojar and others 2016) which includes Europarl v7, Common Crawl, and News Commentary v11 with a total of ca. 4.5 million sentence pairs. The development and testing sets are in this case Newstest2013 and Newstest2014 with 3,000 sentences each.

We report translation quality using BLEU scores over tokenized and truecased texts using the scripts available in the Moses toolkit (Koehn and others 2007).

Setup and Parameters

We use the implementation of the attention-based NMT baseline provided by its authors, dl4mt-tutorial in Python using Theano framework (Theano Development Team 2016). The system implements the attention-based NMT model described previously using one layer of GRUs (Cho et al. 2014). The vocabulary size is 25,000 for English-to-Chinese NMT, and 50,000 for Spanish-to-English and English-German. We use the byte pair encoding (BPE) strategy for out-of-vocabulary words (Sennrich, Haddow, and Birch 2016). For all cases, the maximum sentence length of the training samples is 50, the dimension of the word embeddings is 500, and the dimension of the hidden layers is 1,024. We use dropout with a probability of 0.5 after each layer. The parameters of the models are initialized randomly from a standard normal distribution scaled to a factor of 0.01. The learning rate is 10^{-6} and ρ = 0.95 as the best performance settings in the original paper. The system was trained in 36 epochs on English-to-Chinese, 18 epochs on Spanish-to-English, and 12 on English-to-German. It took 7–12 days of training for each model on a Tesla K40 GPU at the speed of ca. 1,000 words/sec.

Analysis of Results

The BLEU scores of the different NMT models for the three language pairs are shown in Table 1. Along with the NMT baseline, we included an SMT one based on Moses (Koehn and others 2007) with the same training/tuning/test data as the NMT. The results show that all the proposed models outperform the attention-based NMT baseline, and the best scores are obtained by the NMT model with attentive residual connections. Despite their simplicity, the mean residual connections already improve the translation, without increasing the number of parameters.

| Models | BLEU | En–Zh | Es–En | En–De |
|--------|------|-------|-------|-------|
| SMT Winning WMT 14 (2014) | 25.0 | 26.3 | 23.2 |
| NMT Luong, Pham, and Manning (2015) | 26.6 | 25.7 | 22.9 |
| NMT baseline | 22.6 | 25.4 | 22.6 |
| NMT + Mean residual connections | 24.0 | 26.3 | 23.2 |
| NMT + Attentive residual connections | 24.0 | 26.3 | 23.2 |

Table 1: BLEU scores for the evaluated models for three translation directions. The highest score per dataset is marked in bold text. The attentive residual connections makes use of the simple attention function. We show scores reported by other systems for English-to-German on newstest2014.

Role of the attention function.

We now examine the four scoring functions that can be used for attention to residual connections presented above in Eq. 15, considering only English-to-Chinese (the smallest dataset) due to limited computing time.

| Attention function | BLEU |
|--------------------|------|
| Word embeddings, simple | 24.0 |
| Word embeddings + context vector, sum | 23.1 |
| Word embeddings + context vector, product | 23.2 |
| Word embeddings + context vector, dot product | 23.3 |

Table 2: BLEU scores for English-to-Chinese NMT with various attention functions for residual connections on the decoder-side (highest value is in bold).
Performance according to human evaluation.

Manual evaluation on samples of 50 sentences for each language pair helped to ascertain the conclusions obtained from the BLEU scores, and to provide a qualitative understanding of the improvements brought by our model. For each language, we employed one evaluator who was a native speaker of the target language and had good knowledge of the source language. The evaluator ranked three translations of the same source sentence one from each of our models: baseline, mean residual connections, and attentive residual connections according to the translation quality. The three translations were presented in a random order, so that the system that had generated them could not be identified. To integrate the judgments, we proceed in pairs, and count the number of times each system was ranked higher, equal to, or lower than another competing system. The results shown in Table 3 indicate that the attentive model outperforms the one with mean residual connections, and both outperform the baseline, for all three language pairs. The rankings are thus identical to those obtained using BLEU in Table 1.

Table 3: Human evaluation of sentence-level translation quality on three language pairs. We compare the models in pairs, indicating the percentages of sentences that were ranked higher (>), equal to (=), or lower (<) for the first system with respect to the second one. The final ranking shows that the attentive model scores higher than the mean one, and both outperform the NMT baseline.

| System               | En–Zh | Es–En | En–De |
|----------------------|-------|-------|-------|
| Mean vs. Baseline    | > = < | > = < | > = < |
| Attentive vs. Baseline| 26 56 18 | 20 64 16 | 28 58 24 |
| Attentive vs. Mean   | 28 60 12 | 28 56 14 | 32 54 14 |

Table 4: Evaluation of the proposed methods on language modeling. The lowest perplexity (best result) is marked in bold.

| Systems                               | Perplexity |
|---------------------------------------|------------|
| LSTM baseline                         | 88.7       |
| LSTM + Mean residual connections      | 87.1       |
| LSTM + Attentive residual connections | **85.1**   |

Distribution of Attention.

Figure 3 shows the distribution of the attention to previous words obtained with attentive residual connections, on Spanish-to-English NMT (the other two pairs exhibit similar distributions). The weights on the test corpus are summarized using two normalizations. First, we normalize the weights by the number of preceding words, because at each time step the number of preceding words increases, so the weights are inherently more dispersed for words which are at the end of the sentence. Second, we normalize the weights by the sentence length given its variability throughout the test set.

Figure 3a shows the distribution of relative positions which received maximal attention. In other words, we selected, at each word prediction, the preceding word with maximal weight, and counted its relative position, represented on the x-axis. Although the maximal attention was most frequently directed to the closest preceding words, in many cases distant words (more than 30 positions away) were still given maximum attention. Figure 3b shows the mean weight assigned at each word position, now considering the weights assigned to all preceding words (not only

\[ p(y_t | y_1^{t-1}) = g(s_t, d_t) \]  \hfill (18)

For our experiment, we use a Pytorch LM implementation based on the work of [Press and Wolf 2017]. The baseline model is an LSTM with dimension 150 for embeddings and 300 for the hidden states. All the models are trained on the Penn Treebank (Marcus, Marcinkiewicz, and Santorini 1993), for 100 epochs. Table 4 shows the results both proposed methods improving over the baseline. The best performing model uses attentive residual connections and reduces perplexity by 3 points wrt. the baseline.

Qualitative Analysis

We perform a qualitative analysis of the attentive residual connections method. In particular, we analyze the distribution of the weights assigned by the attention mechanism, and the kind of connections that this approach learns. We also present sample translations to contrast the baseline system with the proposed approach.
Figure 2: Examples of Spanish-to-English translations from the attentive residual connections model (target sentences only). The output sentence is on the vertical axis, while the horizontal axis represents the previous words. A darker shade indicates a higher attention weight to each corresponding previous word. The hypothesized syntactic structures are indicated with brackets, and changes of attention focus are visible from the change in shades.

the maximal one as in the previous case). We observe a uniform distribution (with noise for sentences longer than our training samples of 50 words) showing that the attention is indeed distributed throughout a sentence.

Structures Learned by the Attentive Model.

Figure 2 shows matrices with target-side attention weights for four samples translated from Spanish-to-English with our attentive model. The sentence is on the vertical axis, while the horizontal axis represents the previous words. The color shades at each position (“cell”) represent the attention weights. A first attempt to determine classes of words which receive more attention did not lead to conclusive results, except possibly prepositions or conjunctions. In a second attempt, we observed the formation of sub-phrases. By analyzing the changes of attention as a binary tree, it appears on the examined examples that the decoder learns certain syntactic patterns. In Fig. 2 starting from the first word, we mark with brackets the formation of a new sub-phrase each time the focus of attention changes to a new word. In these examples, we can see that the sub-phrases coincide with noun phrases or verb phrases.

Translation Samples. Table 5 shows some examples of translation of the baseline and the attentive residual connections model. The first part (upper three examples) includes cases when the proposed model reached a higher BLEU score than the baseline system. Here, the structure of the sentence or at least the word order is improved. The second part (lower three examples) contains cases where the baseline had better BLEU score than our model. In the first example, the structure of the sentence is different but the information is correct, in the second one the lexical choices differ from the reference but are still correct, while in the third sample, the translation is simply incorrect.

Conclusion

We presented a novel approach to enrich the sequential model of state-of-the-art decoders for neural machine translation. We proposed to extend the context by introducing residual connections to the previous word predictions. To manage the variable lengths of previous predictions, we proposed two methods for context summarization: mean residual connections and attentive residual connections. The first one is an average of the word embedding vectors, which can be interpreted as a representation of the sentence until the time of the new word prediction. The second one uses an attention mechanism over the word embedding vectors to emphasize relevant information from the previous context. We evaluated our proposal over three datasets, for Chinese-to-English, Spanish-to-English, and English-to-German NMT. In each case, we improved the BLEU score compared to the NMT baseline. A manual evaluation over a small set of sentences for each language pair confirmed the improvement. We hypothesized that most of the improvement comes from the language model component, and showed indeed that our model has a lower perplexity than the baseline on a LM
(a) Number of times the word at a given position was assigned with maximum weight. The count is normalized by the length of the sentence and relative word position.

(b) Mean of weights to a given word position. The weights are normalized by the length of the sentence and relative word position.

Figure 3: Distribution of weights to previous words in the attentive residual connections model.

A qualitative analysis showed that the attentive model distributes weights throughout an entire sentence, and learns structures resembling syntactic ones.

To encourage further research in global-context NMT our code will be made available upon publication.

References

[Aharoni and Goldberg 2017] Aharoni, R., and Goldberg, Y. 2017. Towards string-to-tree neural machine translation. In Proc. of the 55th Annual Meeting of the ACL (Vol. 2: Short Papers), 132–140. Vancouver, Canada: Association for Computational Linguistics.

[Bahdanau, Cho, and Bengio 2015] Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations (ICLR).

[Bojar and others 2013] Bojar, O., et al. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In Proceedings of the 8th Workshop on Statistical Machine Translation, 1–44. Sofia, Bulgaria: Association for Computational Linguistics.

[Bojar and others 2016] Bojar, O., et al. 2016. Findings of the 2016 Conference on Machine Translation. In Proceedings of the First Conference on Machine Translation, 131–198. Berlin, Germany: Association for Computational Linguistics.

[Buck, Heafield, and van Ooyen 2014] Buck, C.; Heafield, K.; and van Ooyen, B. 2014. N-gram counts and language models from the common crawl. In Proceedings of the Language Resources and Evaluation Conference.

[Chen et al. 2016] Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; and Jiang, H. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. arXiv preprint arXiv:1609.06038.

[Cheng, Dong, and Lapata 2016] Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 551–561. Austin, Texas: Association for Computational Linguistics.

[Cho et al. 2014] Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734. Doha,
Appendix A: Detailed Architecture

This appendix describes in detail the implementation of the Global-Context NMT model which is based on the attention-based NMT implementation of dl4mt-tutorial.

The input of the model is a source sentence denoted as 1-of-k coded vector, where each element of the sequence corresponds to a word:

\[ x = (x_1, x_2, ..., x_m), x_i \in \mathbb{R}^V \]

and the output is a target sentence denoted as well as 1-of-k coded vector:

\[ y = (y_1, y_2, ..., y_n), y_i \in \mathbb{R}^V \]

where \( V \) is the size of the vocabulary of target and source side, \( m \) and \( n \) are the lengths of the source and target sentences respectively. We omit the bias vectors for simplicity.

Encoder

Each word of the source sentence is embedded in an \( e \)-dimensional vector space using the embedding matrix \( E \in \mathbb{R}^{e \times V} \). The hidden states are \( 2d \)-dimensional vectors modeled by a bi-directional GRU. The forward states \( \overrightarrow{h} = (h_1, ..., h_m) \) are computed as:

\[
\overrightarrow{h}_i = \overrightarrow{z}_i \odot \overrightarrow{h}_{i-1} + (1 - \overrightarrow{z}_i) \odot \overrightarrow{h}'_i
\]

where

\[
\overrightarrow{h}'_i = \tanh(\overrightarrow{W}_E x_i + \overrightarrow{U} \overrightarrow{h}_{i-1})
\]

\[
\overrightarrow{z}_i = \sigma(\overrightarrow{W}_E x_i + \overrightarrow{U} \overrightarrow{h}_{i-1})
\]

Here, \( \overrightarrow{W}, \overrightarrow{W}_E, \overrightarrow{W}_r \in \mathbb{R}^{d \times e} \) and \( \overrightarrow{U}, \overrightarrow{U}_E, \overrightarrow{U}_r \in \mathbb{R}^{d \times d} \) are weight matrices. The backward states \( \overleftarrow{h} = (\overleftarrow{h}_1, ..., \overleftarrow{h}_m) \) are computed in similar manner. The embedding matrix \( \overleftarrow{E} \) is shared for both passes, and the final hidden states are formed by the concatenation of them:

\[
h_i = \begin{bmatrix} \overrightarrow{h}_i \\ \overleftarrow{h}_i \end{bmatrix}
\]

Attention mechanism

The context vector at time \( t \) is calculated by:

\[
c_t = \sum_{i=1}^{m} \alpha_i^t h_i
\]

where

\[
\alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^{m} \exp(e_j^t)}
\]

\[
e_i^t = v^T \tanh(W_d s_{t-1} + W_e h_i)
\]

Here, \( v, W_d \in \mathbb{R}^{d \times e}, W_e \in \mathbb{R}^{d \times d} \) and \( W_e \in \mathbb{R}^{d \times 2d} \) are weight matrices.

Decoder

The input of the decoder are the previous word \( y_{t-1} \) and the context vector \( c_t \), the objective is to predict \( y_t \). The hidden states of the decoder \( s = (s_1, ..., s_n) \) are initialized with the mean of the context vectors:

\[
s_0 = \tanh(W_{init} \frac{1}{m} \sum_{i=1}^{m} c_i)
\]

where \( W_{init} \in \mathbb{R}^{d \times 2d} \) is a weight matrix, \( m \) is the size of the source sentence. The following hidden states are calculated with a GRU conditioned over the context vector at time \( t \) as follows:

\[
s_t = z_t \odot s'_t + (1 - z_t) \odot \tilde{s}_t
\]

where

\[
\tilde{s}_t = \tanh(E y_{t-1} + U \overrightarrow{r}_t \odot \overrightarrow{s}_{t-1} + C \overrightarrow{c_t})
\]

\[
z_t = \sigma(W_t E y_{t-1} + U_z r_{t-1} + C_z c_t)
\]

\[
r_t = \sigma(W_t E y_{t-1} + U_r s_{t-1} + C_r c_t)
\]

Here, \( E \in \mathbb{R}^{e \times V} \) is the embedding matrix for the target language. \( W_r, W_z, W_r \in \mathbb{R}^{d \times e}, W_d, W_e \in \mathbb{R}^{d \times d} \), and \( C, C_z, C_r \in \mathbb{R}^{d \times 2d} \) are weight matrices. The intermediate vector \( s'_t \) is calculated from a simple GRU:

\[
s'_t = GRU(y_{t-1}, s_{t-1})
\]

In the attention-based NMT model, the probability of a target word \( y_t \) is given by:

\[
p(y_t | s_t, y_{t-1}, c_t) = \text{softmax}(W_o \tanh(W_s s_t + W_y y_{t-1} + W_c c_t))
\]

whereas in our global-context NMT model, the probability of a target word \( y_t \) is given by:

\[
p(y_t | s_t, d_t, c_t) = \text{softmax}(W_o \tanh(W_s s_t + W_d d_t + W_c c_t))
\]

Here, \( W_o \in \mathbb{R}^{V \times e}, W_s \in \mathbb{R}^{d \times e}, W_d, W_e \in \mathbb{R}^{d \times e}, W_c \in \mathbb{R}^{d \times 2d} \) are weight matrices. The summary vector \( d_t \) can be calculated in different manners based on previous words \( y_1 \) to \( y_{t-1} \). First, a simple average:

\[
d_t^{avg} = \frac{1}{t-1} \sum_{i=1}^{t-1} y_i
\]

The second, by using an attention mechanism:

\[
d_t^{avg} = \sum_{i=1}^{t-1} \alpha_i^t y_i
\]

\[
\alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^{t-1} \exp(e_j^t)}
\]

The matching function \( e_i^t \) also can be calculated in different manners:

\[
e_i^t = \begin{cases} v^T \tanh(W_y y_i) & \text{simple} \\ v^T \tanh(W_y y_i + W_c c_t) & \text{sum} \\ v^T \tanh(W_y y_i \ast W_c c_t) & \text{product} \\ y_i^T W_c c_t & \text{dot product} \end{cases}
\]

where \( v \in \mathbb{R}^e, W_y \in \mathbb{R}^{e \times e}, \) and \( W_c \in \mathbb{R}^{e \times 2d} \) are weight matrices.

https://github.com/nyu-dl/dl4mt-tutorial