Evaluating Various Attention in Neural Machine Translation

Illy Hoang
University of Connecticut / USA
Illy.Hoang@uconn.edu

***Abstract*—Neural Machine Translation (NMT) quality has significantly improved over a relatively short span of time with the growing number of proposals to address various theoretical limiting factors – one namely being the incorporation of attention layers in model implementations to support contextual readings in predictions during the decoding process. This paper evaluates the translation quality of attention-based NMT with three types of models: 1) a simple seq2seq Encoder-Decoder model that shall be utilized as a 'control' variable for comparisons; 2) a model that computes the attention weights and context vectors of inputted sequences with the additive algorithm proposed by Bahdanau et al.; and 3) a model similar to the former that instead makes computations based on the multiplicative algorithm proposed by Luong et al. The models are trained, validated and tested using randomly generated splits from a data set compiling numerous Spanish-English pair-phrases. They are then evaluated using the BERTScore metric to better assess token similarities between machine translations and their human-produced counterparts. A related repository https://github.com/illydh/nmt-model/tree/main/spr24-research is dedicated to collecting related work.**

I. Introduction and Background

Growing up, my native language was not reinforced on me. I heavily rely on machine translation for communication between myself and non-English language speakers. Through enrollment in foreign language courses growing up, I became fascinated in how humans develop the ability to communicate in various foreign languages and hoped for the opportunity to study this in depth. When I was sought out to create an independent study project, I saw this as my opportunity.

Concurrently taking UConn's course in Artificial Intelligence (AI) piqued my interest in how the subsection of Natural Language Processing (NLP) has allowed the advancement of machine translation by utilizing neural networks (NN) and reinforcement learning methods to train models with the goal of improving NMT performance [14]. I delved into its application to Natural Language Processing (NLP) and its various and numerous contributions through research over time [8]. Although documentation of advancements in NMT research have been extensive, I wanted to take this opportunity to research attention-based models specifically (1) to get a better understanding as to the infrastructure and functionality of NMT models; as well as (2) to model a solution to a presented problem.

Existing research materials proposes new mechanisms for computing attention and display accuracy ratings based on their BLEU Scores, an automated method of human evaluation for NMT [12]. However, due to rapid advancements and improvements to modeling, not much work is documented for comparisons between these models to make true evaluations of their accuracy and effectiveness. Also recent publications argue that NMT evaluation has been too reliant on BLEU as an automatic evaluation metric [11] as well as that the metric poses various implications and inconsistencies in its calculations [3].

In my work thus far, I analyze the composition of a baseline seq2seq model constructed as a recurrent NN as well two other models that includes an attention layer in the decoding process to compute attention weights and to produce context vectors following RL procedures. One method of computation for attention is the additive algorithm proposed by Bahdanau et al. [1] and the other being the multiplicative algorithm proposed by Luong et al. [10]. These machine translations are then evaluated by a BERTScore algorithm which better handles the contextual component of machine translations by (1) reviewing candidate translations token-by-token; and (2) compares the accuracy of these translations against a reference translation, usually a human-judged translation [15].

II. Related Work

A. Neural Machine Translation

Over the better part of the past two decades, NMT has seen a spark in improvements in implementation and performance in models. Stahlberg [13] provides an in-depth review as to how modern NMT implementations use the tokenization and sequencing of sentence phrases to train, validate and test the type of encoder-decoder model later described in section 3.

B. Paper Survey

In this section, we briefly compare this paper to various existing surveys which have reviewed attention methods. As referenced in the previous section, Guo et al. [7] provided a survey of the applications of attention mechanisms in genre of computer visions. Lieskovská et al. [9] reviewed various impacts of recent advancements in speech emotion recognition since the implementation of attention mechanisms in modeling. Ghaffarian et al. [6] investigated the impact of attention mechanisms on deep-learning based remote sensing image processing. Choi et al. [5] proposed their REverse Time AttentIoN (RETAIN) model based on a large Electronic Health Records dataset. Brauwers and Frasincar [2] provided a survey on various attention model implementations unique to specific deep learning models.

III. Proposed Directions of Technical Components

In this section, I will describe the overall architecture of the models for experimentation.

A. Encoder

The initial encoder receives the training input data in the form of padded sequences. The model defines an embedding layer for the encoder to extract meaningful representations from its input sequences. The embedded sequences are then passed through an LSTM cell to compute the outputs from the hidden states at each time step, the final hidden state of the encoder and the context state of the cell.

```
# assuming encoder_input layer is given

encoder_embeddings_layer = layers.Embedding(
                  source_vocab_size,
                  embedding_dim,
                  mask_zero=True,
                  name='encoder_embeddings'
                  )

encoder_embedding_output =
    encoder_embeddings_layers(encoder_inputs)

encoder_lstm_cell = layers.LSTM(
                  hidden_dim,
                  return_state=True,
                  dropout=default_dropout,
                  name='encoder_lstm'
                  )

encoder_outputs, state_h, state_c =
    encoder_lstm_cell(encoder_embedding_output)
```

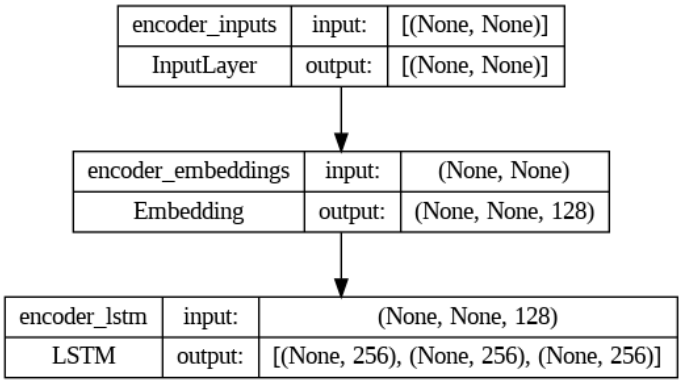

Fig. 1. Visualization of the flow of sequencing in the Encoder

B. Decoder

The decoder is initialized in a similar process where the inputs are its padded training targets, a embedding layer is defined, and an LSTM cell is created. However, when the decoder's embedded inputs are passed through as input for the decoder cell, we pass through the final hidden and context states of the encoder as the initial state of the decoder cell as well to transfer the learned state of the encoder. Lastly, the decoder output sequence, which is the sequence of outputs from the decoder hidden states at each timestamp, is passed through a softmax layer to create a probability distribution of the output word.

```
# decoder_inputs = encoder_outputs for
    no_attention_model
encoder_states = (state_h, state_c)

decoder_embeddings_layer = layers.Embedding(
                  target_vocab_size,
                  embedding_dim,
                  mask_zero=True,
                  name='decoder_embeddings'
                  )

decoder_embedding_output =
    decoder_embeddings_layer(decoder_inputs)

decoder_lstm_cell = layers.LSTM(
                  hidden_dim,
                  return_sequences=True,
                  return_state=True,
                  dropout=default_dropout,
                  name='decoder_lstm'
                  )

#  compute attention to LSTM cell output
    here. Refer to C1) and C2)

decoder_outputs, _, _ = decoder_lstm_cell(
                  decoder_embedding_output,
                  initial_state=encoder_states
                  )

decoder_dense = layers.Dense(
                  target_vocab_size,
                  activation='softmax',
                  name='decoder_dense'
                  )

y_proba = decoder_dense(decoder_outputs)
```

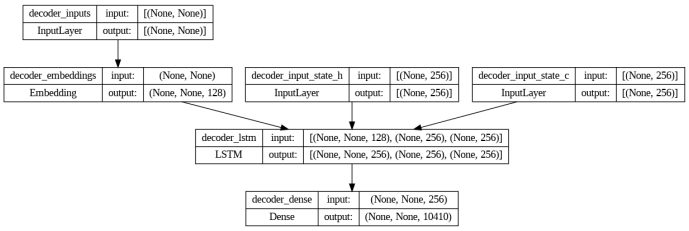

Fig. 2. Visualization of the flow of sequencing in the Decoder

C. Attention Layer

The translations yielded by the proposed model without attention serve as the control of this study. The following schemes for attention mechanisms serve as the test variables of this study:

1) Bahdanau's Attention Mechanism: The additive algorithm proposed by Bahdanau et al. (being referred to as Bahdanau's algorithm moving forward) [1] compiles the text

sequences into numerically-represented vectors to which are padded to a fixed-length. Where attention is computed in the decoder, the following algorithm is called:

```
#  takes encoder_outputs, state_h as input

hidden_extended = tf.expand_dims(state_h,
    axis=1)

#  computing Bahdanau's algorithm
w1 = layers.Dense(hidden_dim,
    name='encoder_outputs_dense1')
w2 = layers.Dense(hidden_dim,
    name='encoder_outputs_dense2')
v = layers.Dense(1,
    name='encoder_outputs_dense')
score = v(tf.nn.tanh(w1(hidden_extended) +
    w2(encoder_outputs)))

weights = tf.nn.softmax(score, axis=1)
context = weights * encoder_outputs
context = tf.reduce_sum(context, axis=1)

#  in decoder
embeddings =
    decoder_embedding_layer(decoder_input)
x_context =
    tf.concat([tf.expand_dims(context,
    axis=1), embeddings], axis=-1)
decoder_output, state_h, state_c =
    self.lstm(x_context)

#  get probability of next sequence
decoder_output_reshaped =
    tf.reshape(decoder_output, shape=(-1,
    decoder_output.shape[2]))
y_proba = decoder_final_layer_dense(
            decoder_output_reshaped)
```

2) Luong's Attention Mechanism: The multiplicative approach proposed by Luong et al. [10] (Luong's algorithm) addresses an issue of information-squashing during training. Where attention is computed in the decoder, the following algorithm is called:

```
#  takes encoder_output, decoder_input as
    input

w = layers.Dense(hidden_dim,
    name='encoder_outputs_dense')
z = w(encoder_output)

#  computing Luong's algorithm
scores = tf.matmul(decoder_output, z,
    transpose_b=True)

weights =
    tf.keras.activations.softmax(scores,
    axis=-1)
context = tf.matmul(attention_weights,
    encoder_output_seq)

#  in decoder
u = tf.keras.layers.Dense(
            hidden_dim, activation='tanh',
            name='attended_outputs_dense'
            )
decoder_output_with_attention = u(
            tf.concat(
               [
                  tf.squeeze(context, 1),
                  tf.squeeze(decoder_output,
                      1)
               ],
            -1))

y_proba = decoder_final_layer_dense(
            decoder_output_with_attention)
```

D. Training

Each model utilizes the Adam optimization algorithm for updating network weights during iterative training. These optimizers are defaulted to a learning rate of 0.001. The control model uses the sparse categorical cross entropy loss function in-place of gradient computations. The test variables use custom loss functions for efficient loss computations during each training step. The test variables also train with a unique algorithm to account for the network weights.

IV. Dataset preparation

The dataset in-use is a .csv file compiling over $100,000$ pairs of English phrases and their translations in Spanish, presumably verified through human judgment. The exact dataset can be downloaded from https://www.kaggle.com/datasets/lonnieqin/englishspanish-translation-dataset. A sample of $95,000$ randomly selected pairs from the set is then split amongst 80% for training, 10% for validation, and 10% for testing.

V. Plan of Experimental Studies by End of Spring

As for the timeline for research, roughly a month was spent on testing that the encoder and decoder were operating to expectation with no attention mechanism initially incorporated in order to both ensure that the implementation of the model can be replicated without direction as well as to allot time for deciding the direction of evaluating these models. The rest of the semester should be spent on either testing other attention models or evaluating existing translation data extracted from models tested in this time frame.

VI. Preliminary Results

Figures 3-6 are diagrams from the notebook of the model with no attention mechanism utilized.

VII. Conclusion

The duration of the semester was focused on compiling the proposed NMT model with various attention mechanisms in order to evaluate their performances. They were incorporated into my proposed mechanism and used to translate a predefined set of 100 Spanish phrases. Their BERTScores are generated using the reference translations for comparison.

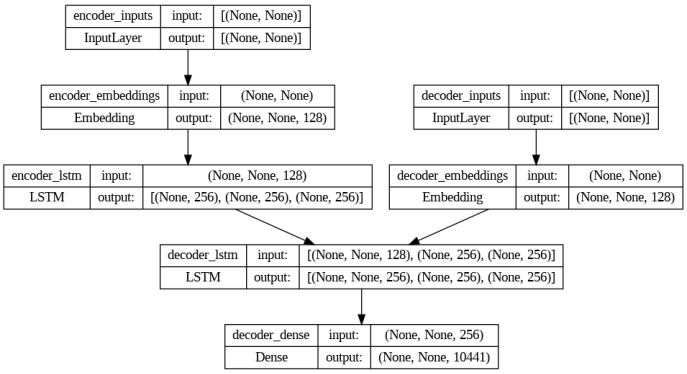

Fig. 3. Compiled model with no attention mechanism

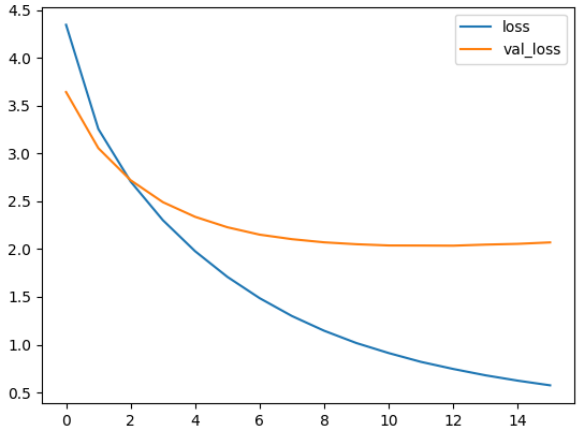

Fig. 4. Loss versus validation loss during model fitting

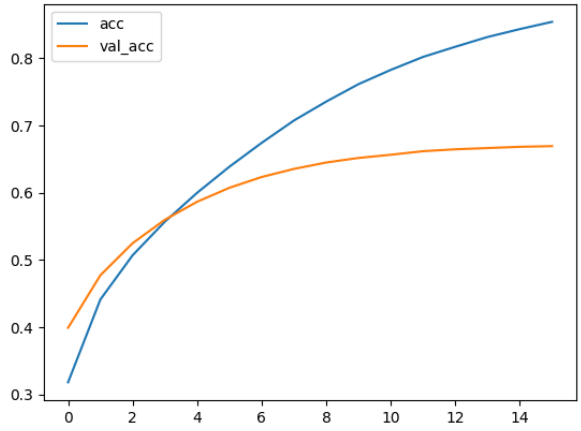

Fig. 5. Model accuracy versus validation accuracy during model fitting

	Tokenized Original	Reference	Translation	Precision	Recall	F1
0	no pueden <unk> .	You can't blame them.	i'm not allowed .	0.4985	0.4664	0.4819
1	el hombre mayor vivia en el apartamento de tre...	The old man lived in the three-room apartment.	the boy lived in three with his car .	0.6373	0.6067	0.6217
2	el me miro fijamente y no dijo nada .	He stared at me and said nothing.	he did his surprise , and i didn't understand .	0.4676	0.5170	0.4911
3	deberias decir la verdad .	You should tell the truth.	you should tell the truth .	1.0000	1.0000	1.0000
4	¿ necesitas mas dinero ?	Do you need more money?	do you need money ?	0.9663	0.8870	0.9249
5	<unk> .	They'll wait.	stop moving .	0.4488	0.3995	0.4227
6	la guerra fria empezo despues de la segunda gu...	The Cold War began after the Second World War.	the war began to get up in 1939 and worse .	0.5773	0.6479	0.6106
7	¿ que clase de <unk> vendes en tu negocio ?	What kinds of goods do you sell in your shop?	what kind of meal do you use in ?	0.7791	0.7298	0.7537
8	¿ tienes la boleta ?	Do you have a receipt?	do you have the same one ?	0.6876	0.7250	0.7058
9	creo que es altamente improbable que vuelva a ...	I think it's highly unlikely that I'll ever se...	i think it's highly unlikely that we'll be bac...	0.6556	0.7261	0.6890

Fig. 6. Evaluation from sample of 10 Spanish phrases translated to English utilizing model with no attention

A. *Final Results*

From Fig. 7, Fig. 8, Fig. 9, the model implemented with Luong's algorithm consistently outperforms the other models, which was expected by its overall intent to solve the bottleneck issue of Bahdanau's algorithm. However, Bahdanau's model seems to under-perform to expectations – including the control. The average scores of the model implemented using Bahdanau's algorithm also varies toa wider range than the other models; while the other algorithms hold a consistent percentage across their respective tables $\pm 1\%$, the percentages of Bahdanau's algorithm have a range of roughly 3%. This may be a discrepancy based on my implementation of the model and shall be amended in future work.

B. *Future Work*

As this paper reaches a rather elusive conclusion, I realize that there is still a lot of research to be done. This was an enriching experience for me to delve into the complicated nature of Machine Learning methods as well as of strong AI software that can emulate accomplishments to complicated tasks such as human translation. While my studies project further into theory, I hope to return to this project and address the issues and concerns that I had initially wished to embark on, namely accuracy. The approach to mathematically marking up textual context as vectors is interesting but even the attention mechanisms experimented with in this study insinuate consistent information loss. Perhaps through some sort of augmentation in data, information can be retained while not decrementing the quality of the model's performance.

A relatively significant step forward in my venture into the genre of NLP would naturally be researching different applications of Large Language Models (LLMs). The survey done by Chang et al. provide a thorough guide of implementation and evaluation of different models as well as modern applications pertaining to LLMs in various fields of study [4]. There are various projects at my disposal for researching LLMs to not just apply but to strengthen my understanding of complex text-generation models that emulate human translation such as generative pre-trained transformers (GPTs) which are in viral use as of late amongst peers.

REFERENCES

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

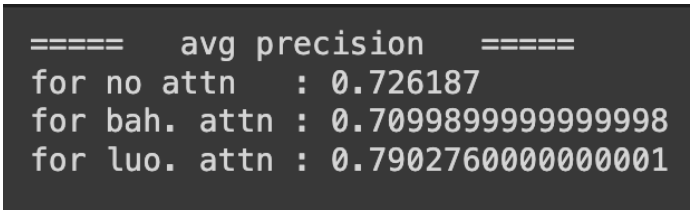

Fig. 7. Average Precision Score across models

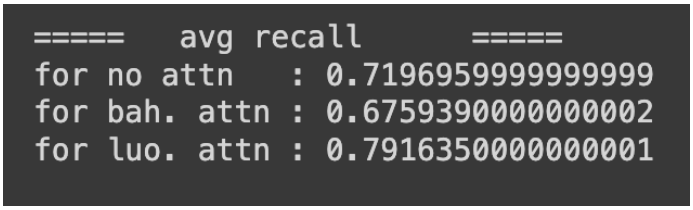

Fig. 8. Average Recall Score across models

[2] Brauwers, Gianni, and Flavius Frasincar. "A general survey on attention mechanisms in deep learning." IEEE Transactions on Knowledge and Data Engineering 35.4 (2021): 3279-3298.
[3] Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. "Re-evaluating the role of BLEU in machine translation research." *11th conference of the european chapter of the association for computational linguistics.* 2006.
[4] Chang, Yupeng, et al. "A survey on evaluation of large language models." ACM Transactions on Intelligent Systems and Technology 15.3 (2024): 1-45.
[5] Choi, Edward, et al. "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism." *Advances in neural information processing systems 29* (2016).
[6] Ghaffarian, Saman, et al. "Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review." *Remote Sensing 13.15* (2021): 2965.
[7] Guo, Meng-Hao, et al. "Attention mechanisms in computer vision: A survey." *Computational visual media 8.3* (2022): 331-368.
[8] Jones, Karen Sparck. "Natural language processing: a historical review." *Current issues in computational linguistics: in honour of Don Walker* (1994): 3-16.
[9] Lieskovská, Eva, et al. "A review on speech emotion recognition using deep learning and attention mechanism." *Electronics 10.10* (2021): 1163.
[10] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).
[11] Mathur, Nitika, Timothy Baldwin, and Trevor Cohn. "Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics." *arXiv preprint arXiv:2006.06264* (2020)
[12] Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting of the Association for Computational Linguistics.* 2002.
[13] Stahlberg, Felix. "Neural machine translation: A review." *Journal of Artificial Intelligence Research 69* (2020): 343-418.
[14] Wu, Lijun, et al. "A study of reinforcement learning for neural machine translation." *arXiv preprint arXiv:1808.08866* (2018).
[15] Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." *arXiv preprint arXiv:1904.09675* (2019).

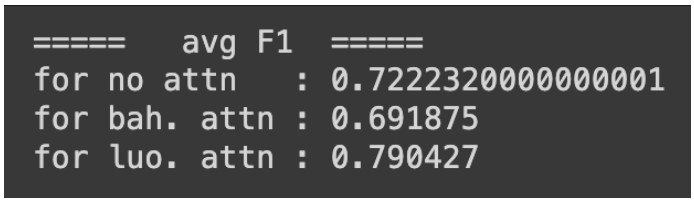

Fig. 9. Average F1 Score across models