Translation vs. Dialogue: A Comparative Analysis of Sequence-to-Sequence Modeling

Wenpeng Hu[1,*], **Ran Le**[2,*], **Bing Liu**[2,†],
Jinwen Ma[1], **Dongyan Zhao**[2], **Rui Yan**[2,‡]
[1] Department of Information Science, Peking University
[2] Wangxuan Institute of Computer Technology, Peking University
{wenpeng.hu, leran, dcsliub, jwma, zhaody, ruiyan}@pku.edu.cn

Abstract

Understanding neural models is a major topic of interest in the deep learning community. In this paper, we propose to interpret a general neural model comparatively. Specifically, we study the sequence-to-sequence (Seq2Seq) model in the contexts of two mainstream NLP tasks–machine translation and dialogue response generation–as they both use the seq2seq model. We investigate how the two tasks are different and how their task difference results in major differences in the behaviors of the resulting translation and dialogue generation systems. This study allows us to make several interesting observations and gain valuable insights, which can be used to help develop better translation and dialogue generation models. To our knowledge, no such comparative study has been done so far.

1 Introduction

The sequence-to-sequence model (seq2seq), especially its enhanced variants of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Bahdanau et al., 2014), have been demonstrated to be highly effective for a variety of NLP tasks, e.g., machine translation (MT) (Sutskever et al., 2014), dialogue response generation (DRG) (Li et al., 2016b), and many others. A seq2seq model takes in a source sentence (a sequence) and generates a target sentence (another sequence). This model was first proposed for MT by translating a source language into a target language. It is now also widely used for DRG to "translate" an input utterance from the user to an output response.

Traditional feature-engineering solutions to MT and DRG require a large number of handcrafted features. In end-to-end learning, seq2seq models require almost no manual features for MT or DRG. The model is purely driven by data: when trained on the translation data, the model learns to translate; when trained on the conversational data, the model learns to converse. However, this approach also makes the neural model hard to understand or interpret. To this end, researchers have proposed to understand and visualize recurrent neural networks (Karpathy et al., 2015) and neural machine translation (MT) models (Ding et al., 2017).

As we know, MT and DRG are two very different tasks. But the same seq2seq modeling works for both, which is counter-intuitive. However, recent studies for understanding neural models have primarily focused on a single task. This paper proposes to study the seq2seq model comparatively in the contexts of two tasks, MT and DRG. Since the two very different tasks use the same model, the internal network behaviors of the model must be very different for the tasks. Our comparative study shows the contrast, which enables us to see a clearer picture of the model for each task and its issues, which cannot be easily observed from the model behaviors of only a single task. This paper aims to answer the following research questions:

1. What are the differences in the network internals and why are they different for the tasks?

[*] Equal contribution
[†] His current affiliation is University of Illinois at Chicago. Email: liub@uic.edu.
[‡] Corresponding author: Rui Yan (ruiyan@pku.edu.cn)

Proceedings of the 28th International Conference on Computational Linguistics, pages 4111–4122
Barcelona, Spain (Online), December 8-13, 2020

2. Which task is harder and why?
3. What network internals have a major impact on the performance of each task?
4. What do we need to do in order to improve the performance of each task?

We answer these important questions starting from analyzing the difference in data for MT and DRG as end-to-end models are data-driven. The objectives of the tasks are embedded in the training data only. We will then study how the different task data lead to different model behaviors in embedding, attention, hidden states, and decoding alignments. These analyses will allow us to answer the above research questions.

2 Background and Related Work

Sequence-to-Sequence model[1]: Given a sequence of inputs $\mathbf{x} = x_1, x_2, ..., x_n$, and a target sequence $\mathbf{y} = y_1, y_2, ..., y_m$, a seq2seq model defines a distribution over outputs ($\mathbf{y}$) and sequentially predicts tokens using a softmax function:

$$p(\mathbf{y}|\mathbf{x};\theta) = \prod_{t=1}^{m} P(y_t|\mathbf{x}, \mathbf{y}_{<t};\theta) = \prod_{t=1}^{m} \text{softmax}(f(\mathbf{h}_t, \mathbf{c}_t, y_{t-1})) \quad (1)$$

where $f(\cdot)$ is a non-linear function, y_{t-1} is the generated previous word/token obtained from a word look-up table, $\mathbf{h}_t = \text{LSTM}(y_{t-1}, \mathbf{h}_{t-1})$ is the hidden state variable of the LSTM network at time step t, $\mathbf{h}_{t-1}$ is the previous hidden state of the LSTM network, $\mathbf{c}_t = \sum_{i=1}^{n} a_{i,t}\mathbf{b}_i$ is the attention-based encoding (Bahdanau et al., 2014) of $\mathbf{x}$ at decoding time step t, $\mathbf{b}_i$ is the decoder hidden state at time step t and it has the same computational formulation (but different parameters) as $\mathbf{h}_t$, and $a_{i,t}$ is an attention weight calculated by Eq. (2).

The seq2seq model has been shown to have excellent performance for both MT (Koehn, 2017) and DRG (Serban et al., 2017). However, to understand how and why it is effective on different tasks remains to be a challenge.

Understanding Neural Models: Visualization techniques have been explored in computer vision to understand neural networks (Simonyan et al., 2013; Nguyen et al., 2015; Vondrick et al., 2013; Szegedy et al., 2013; Mahendran and Vedaldi, 2015; Zeiler and Fergus, 2014). For visualization methods in NLP, a few ablation studies have analyzed the effects on performance of several internal neural units in specific NLP tasks (Ding et al., 2017; Li et al., 2016a; Shi et al., 2016). The earliest visualized neural unit was the word embedding, which projects the word embedding space into a 2-dimensional space and observes that words with similar meaning tend to cluster together (Ji and Eisenstein, 2014). Li et al. (2016a) described strategies for visualizing compositionality in neural models (mainly focused on some specific NLP tasks such as sentence classification), including the first-order derivatives and the variances. Ding et al. (2017) focused on MT and proposed to use layer-wise relevance propagation to compute the contribution of each contextual word to arbitrary hidden states in the attention-based encoder-decoder framework.

Unlike previous works, this paper performs a comparative study of the seq2seq model for MT and DRG. To our knowledge, this has not been done so far. It analyzes and visualizes different behaviors of the seq2seq model for MT and DRG to answer the proposed research questions.

3 Datasets Analysis

We start our investigation from the most direct source of difference: the characteristics of the data for MT and DRG. The following datasets are used in our analysis.

MT Dataset: For MT, our training dataset consists of 2.08M sentence translation pairs in different languages, extracted from the LDC corpus. We use NIST 2003 Chinese-English dataset as the validation set, and NIST 2004-2006 datasets as test sets. Each NIST dataset has 4 different reference translations for each sentence, and we also call it the Multi-Reference Dataset (MRD).

[1]The LSTM and GRU units usually have similar performance (Greff et al., 2017). Without loss of generality, we illustrate using the LSTM-based seq2seq model.

Dialogue Dataset: We use the twitter dataset (Ritter et al., 2011) for English and a question and answer dataset about movies (Wu et al., 2016) for Chinese which has 1.61M sentence pairs. We randomly selected 2k sentence pairs from the corpus for testing and another 2k sentence pairs for development. To analyze the diversity in replies, we asked 4 Ph.D. students to reply 300 queries in any way they see fit and use their replies as the multi-references for dialogues.

3.1 Diversity Analysis

Diversity is a common phenomenon in both MT and DRG. One can always reply to or translate a given sentence in different ways. Here, we analyze the diversity phenomenon and its influence on the automatic evaluation metrics like BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE-L (Lin, 2004), and Diversity (diversity of a set) (Zhang and Hurley, 2008). In the experiment, *we alternately choose one standard answer from the multi-references (the test dataset has several ground-truth target translations or replies for each source sentence) as the test sample and the rest of the standard answers as its references (similar to K-fold cross validation).* We average the scores from all references.

From Table 1, we can see that the dialogue data scores much lower (column 2-4) than the translation data. Since all standard answers are written by humans, we can conclude that the diversity of dialogues is much higher than that of translation, which is also reflected by the much higher diversity score of dialogue than translation. It is interesting to see that even the ground truth results do not get high scores in automatic evaluations. This indicates that the diversity phenomenon has a great influence on automatic evaluation metrics.

	METEOR	ROUGE	BLEU	Diversity
Translation	0.32	0.54	42.63	0.66
Dialogue	0.08	0.31	8.59	1.07

Table 1: Evaluation results using the ground-truth data of translation and dialogue.

Apart from *high diversity* of dialogue responses, a related data difference for MT and for DRGs is that there is a high correlation between input length and output length for translations (please see Appendix A.1), but this is not the case for dialogue responses, which again indicates *high diversity* of dialogues. With these data differences in mind, we next study the model differences.

4 Ability of Training Embedding

Word embedding plays a vital role in seq2seq models. Considering the decoding process, it is hard for the decoder to predict the correct words if word embeddings are unable to cluster similar words together. Here, we compare the word embedding quality of MT and DRG dased on *similarity* and *analogy*.

Training Details: For a fair comparison, we employ the same open-source seq2seq platform[2] (Klein et al., 2017) to train both the MT and DRG models. The hyper-parameters in our system are described as follows. We limit the vocabulary to 30k in our experiments. The size of hidden units is 600 and the word embedding dimension is 300. We set the number of layers of LSTM to 2 in both encoder (bi-directional encoder is adopted) and decoder. The network parameters are updated using the Adam algorithm (Kingma and Ba, 2014) with learning rate of 0.0001. For the final decoding at test time, we adopt the beam search with beam size b=10 for MT and b=3 for DRG. A large beam size for dialogues tends to generate trivial or universal responses like "I don't know". The evaluation results of the seq2seq model applied in the dialogue experiment achieves 9.75, 1.06, 0.14, 0.043 for BLEU 1 to 4 respectively, which are better than the results reported in (Wu et al., 2017).

4.1 Word Similarity Analysis

Word similarity reflects the semantic performance of word embeddings (Mikolov et al., 2013a). We use the WordSim-353 dataset (Finkelstein et al., 2001) to evaluate the word embeddings generated by the seq2seq model for the two different tasks. WordSim-353 is a standard dataset for evaluating English word embeddings, which contains manually compiled sets of similar/synonymous words. We hired three

[2]https://github.com/OpenMT/OpenMT-py

Ph.D. students to collaboratively translate the WordSim-353 dataset into Chinese as the Chinese word similarity evaluation dataset.

Table 2 shows MT word embeddings achieve better similarity results. We believe the main reason is that multiple references (responses) for a source sentence in dialogue are clustered together but words in these references are not necessarily similar due to the high diversity in the DRG data.

4.2 Word Analogy Analysis

The word analogy is a word game, e.g., using the calculated embedding of "Queen - King + Man" to infer "Woman", introduced by (Mikolov et al., 2013a) for evaluating word embeddings' grammatical performance. We employ the word2vec toolkit (which also include datasets) provided by Mikolov et al. (2013b) to calculate the analogy score. For the Chinese datasets, we adopt the same processing method as in the word similarity task, i.e., translating the English datasets into Chinese.

	Encoder		Decoder	
	Chinese	English	Chinese	English
DRG	20.36	39.10	13.42	30.75
MT	37.45	49.43	38.98	46.57

Table 2: Experimental results on the WordSim-353 dataset. The numbers in the table are Spearmans correlation recorded as $\rho \times 100$ between the embedding similarity and human judgments.

Table 3 shows the word analogy evaluation results which give the same conclusion as the word similarity evaluation. Another interesting phenomenon that can be observed from both the word similarity and analogy evaluations is that the word embedding quality is related to the direction of the translation in MT. For instance, the word embedding trained by English-to-Chinese translation has a higher quality than that trained by Chinese-to-English translation. For monolingual dialogues, there is no such observation.

5 Analyzing Attention and Hidden States

In addition to embeddings, attention is also an important mechanism for seq2seq models.

5.1 Analyzing Attention Distributions

The attention mechanism (Bahdanau et al., 2014) is an approach to dynamically determine the relevant source context for each target word. The attention weight $a_{i,t}$ used in Eq. (1) indicates how well the source word x_i and the target word y_t are matched and it is computed by:

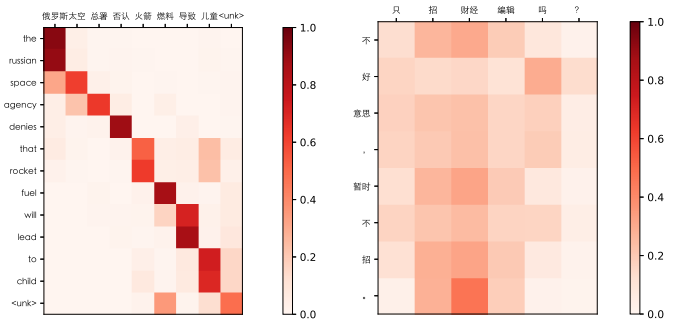
	Encoder		Decoder	
	Chinese	English	Chinese	English
DRG	9.31	14.67	10.09	20.88
MT	11.96	27.29	13.02	24.52

Table 3: Experimental results of the word analogy task measured in accuracy (shown as percentage).

$$a_{i,t} = \frac{\exp(e_{i,t})}{\sum_{i=1}^{m} \exp(e_{i,t})} \quad (2)$$

$e_{i,t} = \mathbf{h}_t \mathbf{W}_a \mathbf{b}_i$ scores the match for $\mathbf{h}_t$ and $\mathbf{b}_i$.

Fig. 1 gives the heat map of the attention matrices. We can observe that each target word in MT can be soft-aligned very well with a source word that has similar meaning to it. That is, there is a clear approximately *one-to-one correspondence* relation. However, all the target words in DRG tend to be soft-aligned with about the same subset of the words in the source sentence, and the alignments are much weaker. This implies that the generated responses only focus on only a few words in the source sentence through the attention mechanism. To



(a) (b)

Figure 1: Heat map of the attention weight matrices for MT 1(a) and for DRG 1(b). The sentence displayed along the horizontal axis is the source language sentence in MT or the query in DRG. The sentence displayed along the vertical axis is its corresponding translation or response.

measure this phenomenon, we adopt the KL-Divergence (KL) to gauge how spread the attentions are on the source words:

$$\text{Div}^a = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{m_j}\sum_{t=1}^{m_j}\text{KL}(p_{j,t}||\tilde{p_j}) \tag{3}$$

where $\tilde{p_j} = \frac{1}{m_j}\sum_{t=1}^{m_j} p_{j,t}$; j indicates the j-th instance in the test dataset (which contains 2k (N) sentence pairs); $p_{j,t} = (a^j_{1,t}, a^j_{2,t}, ..., a^j_{n_j,t})$, n_j is the source sentence length and m_j is the generated sentence length, and $a^j_{i,t}$ is the attention weights calculated by Eq. (2).

Table 4 shows a large difference between DRG and MT in terms of the spread of attentions over all source words. Clearly, in DRG, all words of a reply focus on roughly the same subset of words in the source. Intuitively, it means different replies respond to the same aspects (i.e. the attention-concentrated words) of the input utterance.

	DRG	MT
Div^a	0.036	1.202

Table 4: Statistical results on the spread of attentions on the source.

5.2 Analyzing Hidden States

Due to the high-diversity of DRG, each response attends to a portion of input utterance. We hypothesize that the encoded information (hidden states) of dialogues may be lack of diversity. As for translation, all words are attended to and the encoded information is richer.

To verify this hypothesis, we first select one sentence randomly as the test sentence (in translation and also in dialogue) to extract the hidden states of MT and DRG. Then we employ the PCA to project these hidden states into two dimensions and show them in Fig. 2.

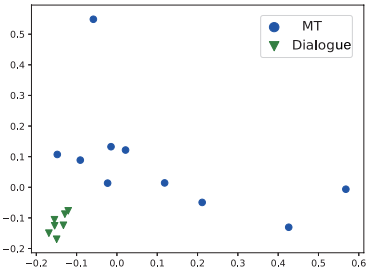


Figure 2: Scatter plot of decoder hidden states

From Fig. 2, we can see that the hidden states of dialogues tend to be clustered together which means they have small differences. We also give the statistical results on 2k sentence pairs to avoid the impact of outliers. The statistical results (see Table 5) reach the same conclusion.

It is interesting to see that when all attentions are concentrated, encoded states are also concentrated. When attentions are spread, hidden states are dissimilar. Dialogues concentrate on certain words while translations pay attention to the whole source sentence.

6 Analyzing the Final Prediction

In seq2seq models, the *attention-based* encoding and the decoder hidden states are two important constituent parts for making the final prediction. Exploring their contributions to the final decision can help NLP researchers to design better models. We propose a new *combinational disturbance* approach to measuring how much each input unit contributes to the final decision. First, we need to introduce the *first-derivative salience* method (Li et al., 2016a).

First-order Derivatives (FD): The final decision of a seq2seq model is determined by the decoder output distribution $\varphi(\mathbf{u})$, which is computed using the softmax function over all the words in the vocabulary V. Let $\mathbf{u}$ be an internal neural unit and it can be seen as an input unit for the final decision making.

	DRG	MT		DRG	MT
Var	5.73	9.13	**Sim**	0.730	0.496

Table 5: Statistical results for decoder hidden states. 'Var' is the average variance for each dimension in the states. 'Sim' is the average of the similarities among the states.

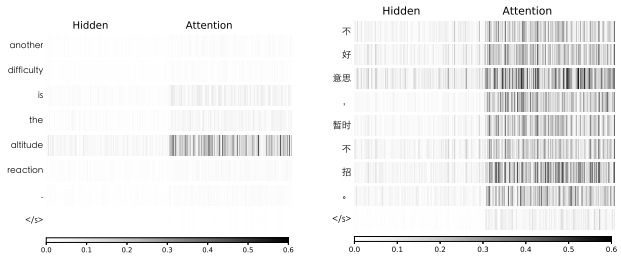
We denote the probability of each possible output word v ($\in V$) as $\varphi_v(\mathbf{u})$. $\varphi_v(\mathbf{u})$ can be approximated with a linear function of $\mathbf{u}$:

$$\varphi_v(\mathbf{u}) \approx w_v(\mathbf{u})^T\mathbf{u} + b \tag{4}$$

where $w_v(\mathbf{u}) = \frac{\partial(\varphi_v)}{\partial \mathbf{u}}|_{\mathbf{u}}$. $|w_v(\mathbf{u})|$ indicates the sensitivity of the final decision to any change in each dimension, which tells us how much each dimension of the input neural unit contributes to the final decision.

Fig. 3 shows the results. We can see that attention contributes more than hidden states to the final predictions, especially for DRG. On the other hand, the first-order derivatives for MT are much smaller than those for DRG which may indicate that MT has a better anti-interference ability. To evaluate the total contributions of the input neural unit, we compute the norm of these derivatives and the results are given in Fig. 4.

Combinational Disturbance (CD): The first-derivative salience approach measures how much each dimension of the input neural unit contributes to the final decision. However, it is limited when employing it to measure the whole contribution of an input neural unit to the final decision. This problem will not be alleviated even using the norm of first-derivative vector. That is because we can only get the derivative from one direction but not others. To tackle this problem, we propose to use the degree of change to the final output distribution brought by adding tiny random disturbances δ to the input unit as its contributions:

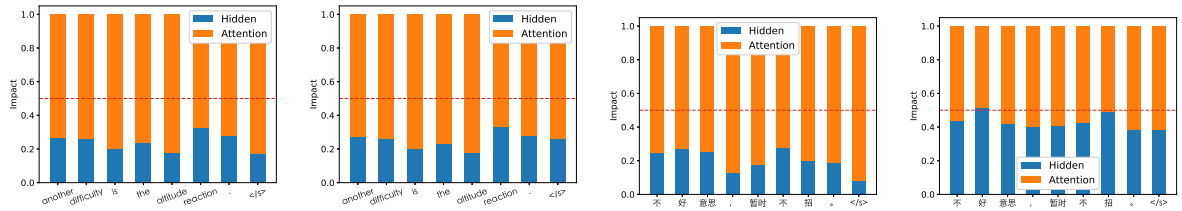


(a) (b)

Figure 3: Salience heat map of the attention and hidden state, 3(a) for MT and 3(b) for DRG.

$$w_v(\mathbf{u}) = \frac{1}{N}\sum_{i=1}^{N}|\varphi_v(\mathbf{u}+\delta_i) - \varphi_v(\mathbf{u})| \tag{5}$$

δ_i is the tiny random disturbance randomly produced from $[-10^{-4}, 10^{-4}]$ in the experiment.



(a) MT (b) MT (c) DRG (d) DRG

Figure 4: (a) (c): Total contributions of attention and hidden state calculated by First-order Derivatives; (b) (d): Total contributions of attention and hidden state calculated by Combinational Disturbance.

The figure (b) and (d) in Fig. 4 shows the results obtained by our method. Compared to the results of using FD, the contribution of hidden states has increased in DRG, while it remains unchanged for MT. We believe that this is because in MT, the source input provides a "strong supervision" to align the target with references, e.g., co-occurrence frequency, while for DRG, there is no strong supervision from references (given responses). Therefore, the effect of language models is emphasized.

7 Decoding Confidence

The entropy of decoder output distribution φ (described in Section 6) can be regarded as a representation of how confident the model is at the current decoding time step t. It can be calculated by $S(\varphi_t) = \sum_l -\varphi_{t,v}\ln(\varphi_{t,v})$. v denotes the v-th word in the vocabulary. In this case, smaller entropy means more confidence in decoding.

Case comparison. We first explore the decoding confidence at each decoding position for both MT and DRG systems. Fig. 5 shows the entropy heat map for six cases randomly sampled from the test sets, from where, we can make the following observations:

• The entropy of MT is generally lower than that of DRG . That is, the decoder is more confident in MT due to the strong supervision from the inputs and the nature of the task. Due to the high diversity of DRG without strong supervision, the decoder in DRG is not very confident in decoding.

• Both systems seem rather confused at the first decoding step (with high entropy values). One possible reason for this is that the decoder has several possible ways to translate or to respond the sentence encoded by the encoder (the internal vector representation). It is difficult to decide what to decode to at the first decoding step for both tasks.

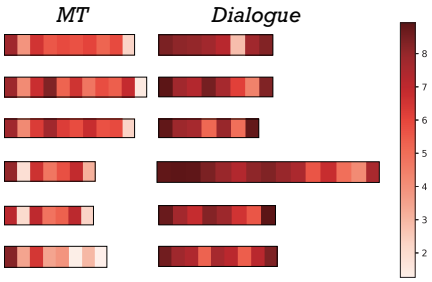


Figure 5: Entropy of each decoding step (one block for one step/word) for 6 cases sampled from test sets.

• MT has a higher confidence in deciding when to terminate the decoding process. It is intuitive to finish decoding when the length of the target sentence is comparable to that of the source sentence. Besides, translations are strongly supervised: when the sentence is completely translated, the decoding process should end. DRG has little guidance. The decoder has difficulty to decide when to stop.

Overall comparison: We now use 2k test sentences to give the statistics about the above observations. We compute the average entropy $\tilde{S}_t$ at each word position with the following formula:

$$\tilde{S}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} S(\varphi_t^i) \tag{6}$$

where N_t denotes the number of words at the t-th decoding position. Fig. 6 visualizes the distribution of the average entropy for each position. Detailed confidences are given for the first 30 decoding steps as only a small fraction of samples will not stop decoding (the decoding length over 30, 24.9% for the MT system, 0.7% for the DRG system). We can see that both systems are uncertain about what to generate at the beginning. The entropy reduces sharply as the decoding process moves forward, and then remains stable. We also see that MT has more confidence in the entire decoding process.

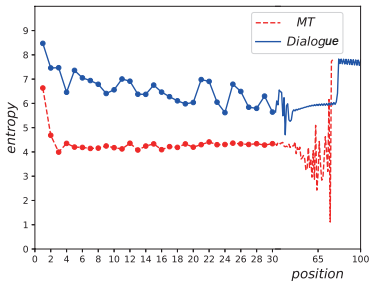


Figure 6: Average entropy of each decoding step for test set.

To further verify the observations, we select sentences with fixed lengths to decode, and compute average entropy at each position. The same phenomenons are observed in Fig. 7.

'EOS' Generation. Here we focus on only the 'EOS' token and study how confident the two systems are when they decide to end the decoding process. Fig. 8 shows that compared the entropy values in Fig. 6, the gap of end decoding confidence between the two systems is further widened. For DRG, it is hard for the model to decide whether to end the decoding process at every position, with the entropy as large as 8. For MT, although the entropy of 'EOS' token is large for short sentences (shorter sentences are usually harder to translate), it lows down as the length increases. MT can make more confident decisions at the positions ranged from 13 to 30.

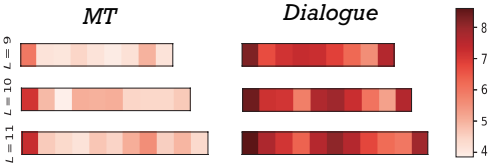


Figure 7: Average entropy of each decoding step for sequences with a fixed length. L-9/L-10/L-11 denote length of 9, 10, 11, respectively.

8 Error Analysis

This section investigates the conditions that a correct prediction is likely to be made by the model. We use negative log-likelihood (NLL) as the metric to measure the correctness of the model's prediction. NLL is a negative log function of the prediction probability of ground-truth and it can be calculated by $s_i = -\log(p_i)$. In our case, p_i is the prediction probability for the i-th correct word in the given replies (or translations) predicted by the seq2seq model. To alleviate the influence of the high diversity

problem discussed before, we use teacher-forcing (Williams and Zipser, 1989) in this experiment. A good prediction has a small NLL. As for the sentence level measurement s, we average all the s_i in the sentence, that is, $\overline{s} = \frac{1}{L}\sum_{i=1}^{L} s_i$, where L is the sentence length.

Extensive experiments have been conducted to investigate many possible factors that may cause the model to make mistakes (or higher NLL), including: the position of the decoding word, the average of hidden state change volume over the whole prediction, word frequency, attention distribution (introduced in Section 5), and the length of generation. We found that word frequency and attention distribution have clear correlation with the correctness of prediction (NLL). The rest of the factors show little correlation.

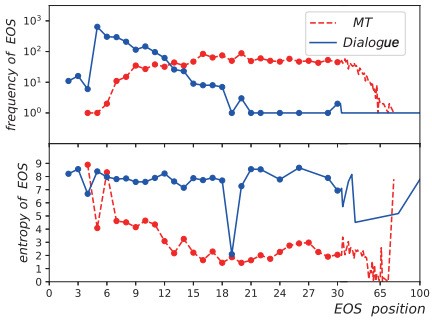


Figure 8: Average entropy in deciding 'EOS' and frequency of 'EOS' generation at each decoding position. Sharp fluctuations appear when the sentence length is small.

Fig. 9 shows the correlation between the word frequency and NLL of the DRG system and MT system. We can see that both MT and DRG systems have a tendency to make mistakes on low frequency words and perform well on high frequency words. This phenomenon indicates that words with higher frequency are trained better by the model, yielding better performances. This phenomenon can also be regard as an imbalanced prediction problem in DRG and MT. Thus, balancing or increasing word frequency can help the model get better results.

Pearson coefficient and Spearman coefficient are calculated to measure the degree of correlation between word frequency and NLL. Their results are given at the top of Fig. 9. We can see a higher correlation between word frequency and NLL scores in DRG. That is because the high diversity phenomenon of DRGs such that it is harder for the seq2seq model to capture the correspondence between the source and target when the word frequency is low. As a result, DRG is more sensitive to word frequency than MT.

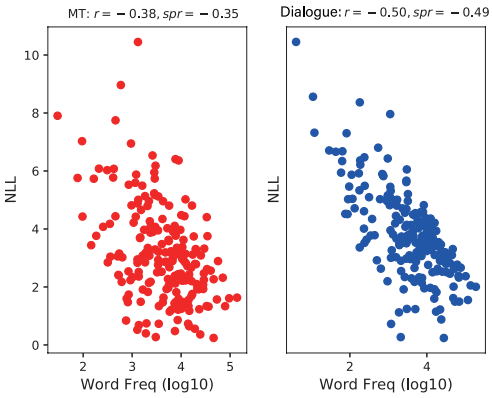


Figure 9: Correlation between word frequency and NLL. Each point denotes a word. (MT is on the left and DRG is on the right).

In Section 5, we analyzed the attention distribution. Here we found that attention distribution can clearly indicate the accuracy of the predicted words or sentences in MT. Similar to but also different from Section 5, here we use entropy to measure the attention dispersion of each word/step over the source language inputs. Intuitively, a larger entropy corresponds to a more dispersed attention. We use the average of the entropy values at all decoding steps as the measure of the *degree of attention concentration* of the whole sentence.

Fig. 10 shows the correlation between the entropy (degree of attention concentration) and NLL (which reflects the quality of the generated target sentence). It can be easily observed that the performance of MT is closely related to the degree of attention concentration. The Pearson coefficient and Spearman coefficient shown in the figure are much higher than those coefficients for DRG. We can see that when the attention is focused (i.e., the weight gathers on a few tokens) the model has less tendency to make mistakes. Errors are more likely to occur when the attention mechanism cannot generate a focused weight distribution. While for DRG, the performance is not closely related to the degree of attention concentration for both Pearson and Spearman coefficients, smaller than 0.03. The reason for the difference of the two systems or applications is that there is a clear word alignment relationship in MT, which is not the case for DRG. The word alignment relationship provides a strong supervision for the decoding step

in translation. A dispersed attention indicates that the current prediction has a high-level of uncertainty and thus tends to make mistakes. Due to the high data diversity problem, DRG lacks a clear alignment relationship between source and target, and thus the degree of attention concentration cannot reflect the quality of the generated target sentence.

9 Summary of the Comparative Study

We now summarize the study by answering the four proposed research questions.

1. *What are the differences in the network internals and why are they different for the tasks?* The answer is that for all the analyzed items, the network internals behavior differently for different tasks. The reason is due to the data difference of the two tasks, which reflects the nature of the two tasks. The possible responses for an input query in DRG can be highly diverse, much more than possible translations of a sentence in MT.

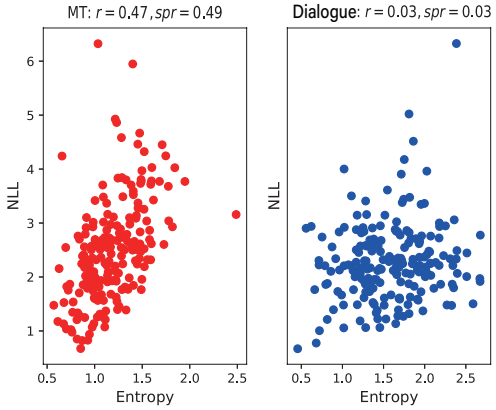


Figure 10: Correlation between the degree of attention concentration and NLL. Each point denotes a sentence. MT is on the left and DRG is on the right.

2. *Which task is harder and why?* Due to *high diversity* in DRG, DRG appears to be a harder problem because its decoder is less confident on what word to generate, and when to stop because of the uncertainty in attention and no length guidance from the source sentence. DRG's attention is more smeared rather than focused like MT (which has a high degree of attention concentration). MT does much better on these, which enables a MT system to more confidently generate a translation.

3. *What network internals have a major impact on the performance of each task?* For different tasks, we have different answers. For MT, attention has a great impact on the final results. More focused attention means better translations. But for DRG, the answer is unclear. Its attention distribution does not have a strong correlation with the correct output. For both tasks, word frequency is a major factor that influences the performance.

4. *What do we need to do in order to improve the performance of each task?* For translation, since the degree of attention concentration has a clear correlation with the translation quality, it may be used as a translation quality measure. In designing new MT algorithms, we should try to improve the attention mechanism to enable it to have a higher degree of attention concentration. Furthermore, as more frequent words are more likely to be translated well, in data collection, one should focus on collecting more data containing those less frequent words. This is also the case for DRG. But overall, DRG appears to be a less well understood problem. The seq2seq model and attention may not be sufficient for the task.

10 Conclusions

This paper proposed to interpret the seq2seq model by comparing its behaviors in the contexts of machine translation (MT) and dialogue response generation (DRG). We analyzed the model from multiple perspectives and showed several model differences for the two tasks. Our work included comparisons of data distribution, word embedding, attention mechanism, state prediction, decoding confidence, and when and where errors tend to occur. The analysis led to some interesting observations and valuable insights. We believe more such analyses should be conducted in the future to other models and tasks to guide researchers in designing better algorithms.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China(NSFC No. 61876196). This work is supported by Beijing Academy of Artificial Intelligence(BAAI).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *ACL*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20:116–131.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutnx00EDk, Bas R. Steunebrink, and J Schmidhuber. 2017. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2222–2232.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9 8:1735–80.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *ACL*.

Andrej Karpathy, Justin Johnson, and Fei fei Li. 2015. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Klein, Yoon Kyung Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *ACL*.

Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Daniel Jurafsky. 2016a. Visualizing and understanding neural models in nlp. In *HLT-NAACL*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016b. A persona-based neural conversation model. *CoRR*, abs/1603.06155.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries.

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

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

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436.

Kishore Papineni, Salim E. Roucos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *EMNLP*.

Iulian Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *EMNLP*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. 2013. Hoggles: Visualizing object detection features. *2013 IEEE International Conference on Computer Vision*, pages 1–8.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2016. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots.

Yu Wu, Wei Wu, Dejian Yang, Can Xu, Zhoujun Li, and Ming Zhou. 2017. Neural response generation with dynamic vocabularies. *CoRR*, abs/1711.11191.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Mi Zhang and Neil J. Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*.

A Data Analysis

A.1 Data Length Analysis

Target sequence length is an important piece of information for the decoder in the seq2seq model to generate translations (in MT) or responses (in DRG). This section, discusses the correlation analysis of the sequence length between the source and target for MT and DRG. We use NIST 03-06 as our MT experimental set. For DRG, we employ the human utterances as the experimental set. Since there are multiple responses for each test sequence, we use the average length as the target sequence length.

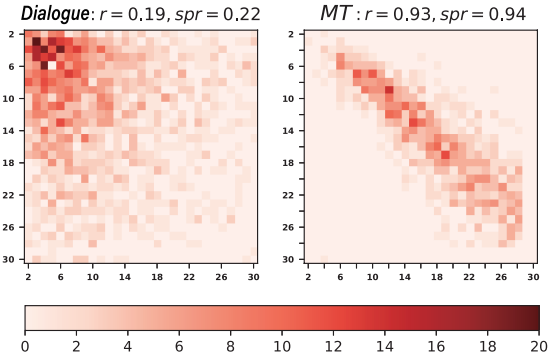


Figure 11: The joint frequency for sequence length between source and target. The horizontal ordinate represents source sequence length. 'r' denotes Pearson correlation coefficient; 'spr' denotes the Spearman correlation coefficient.

Fig. 11 shows the results. We can observe that the target sequence length in the MT corpus is highly correlated with the source sequence length (indicated by high correlation coefficient and diagonal distribution of length). However correlations for DRG are scattered, which affects the network's control over the decoding end time (see Section 7 for more details).