Natural Language Understanding (Spring ’17) – Task 2: Dialogue

Group 21: Sara Javadzadeh, Lierni Sestorain, Georgios Touloupas, Qin Wang
Department of Computer Science, ETH Zurich

I. Introduction

In this report, we explain how our dialogue system is built. To develop our model, we use Google’s tf-seq2seq library [1, 2] for Tensorflow [3], implementing seq2seq [4] architectures.

As our baseline, we use a basic seq2seq model with bidirectional RNN encoder, whose last state is passed as the initial state of the basic decoder. However, given the poor results of the baseline, we try many approaches to improve our model.

Firstly, we increase the training data by 50% by merging the Cornell Movie-Dialogs Corpus [5] with the given MovieTriples dataset [6]. In addition, we try the Bahdanau attention [7] and Dot attention [8] mechanisms. Moreover, we experiment with residual connections on the encoder and decoder, as well as using a convolutional encoder. Finally, we improve the results of our best model by exploiting the three-turn structure of the dataset and increasing the size and depth of the model.

In Section II, we explain in detail the network architecture of each model we experimented with. Section III includes a description of the preprocessing performed on the dataset, a presentation of the metrics used for evaluation, a presentation of the results of our experiments and a quantitative and qualitative comparison of the corresponding models. This analysis leads to a final model which converges to a 3.956 loss value after just 5K steps (2.5 hours of training) and achieves 0.85 BLEU score and 4.24 average perplexity, summed up in Section V.

II. Network Architecture

A. Baseline Model

Our baseline is a basic seq2Seq model which uses no attention mechanism. Its encoder is a bidirectional RNN encoder with multiple layers. The last encoder state is passed through a fully connected layer and used to initialize the decoder which has also multiple layers. The model uses LSTM cells.

B. Attention Mechanism

We use an RNN decoder that uses attention over the input sequence. We compare two different attention mechanisms, namely, Bahdanau [7] and Dot attention [8]. In a nutshell, the Bahdanau mechanism uses an attention layer that calculates attention scores using a parameterized multiplication while the Dot mechanism calculates them using a dot product.

C. Convolutional Encoder

We also experiment with a convolutional encoder model [9]. Instead of a bidirectional RNN encoder, we use an architecture based on a succession of convolutional layers. This allows to encode the entire source sentence simultaneously, in contrast to RNN networks, leading to faster training time.

D. Residual Connections

We also experiment with residual connections [10, 11] between the layers of the encoder and the decoder. Residual connections improve the gradient flow during backpropagation, which enables us to train deeper models faster, achieving better results than shallower models. We use dense residual connections [12], where every layer of the encoder and decoder is connected to every previous layer.

III. Experiments

A. Datasets

1) MovieTriples dataset: The MovieTriples dataset [6] was provided to use for training and validating our models. The dataset contains conversation triples from movie scripts. The triples have the form A-B-A, where A is making an utterance followed by B making an utterance followed by A again. We construct the training and validation datasets by splitting each triple into two source-target utterance pairs (utterances 1-2 and 2-3).

2) Cornell Movie-Dialogs Corpus: To increase the size of the training set, we merged the MovieTriples dataset with the Cornell Movie-Dialogs Corpus (CMC) [5]. Before merging, we preprocessed the CMC the same way the MovieTriples dataset have been preprocessed. Using the natural language toolkit NLTK [13], the dataset was tokenized and named-entity recognition was applied to replace all names and numbers with `<person>` and `<number>` tags respectively. In CMC, conversations can have two or more utterances, so we construct as many source-target utterance pairs as possible (1 pair for 2 utterances, 2 for 3 etc). The merged MovieTriples-CMC dataset has 50% more pairs than the original.

3) Exploiting the Three-Turn Structure: To exploit the triple nature of the MovieTriples dataset, we make use of the first utterance when predicting the third, in contrast to using only the second. To do so, we construct additional utterance pairs where the source is utterances 1 and 2 concatenated (separated with a new `<utterer_change>` token) and the target is utterance 3. Regarding the CMC, where conversations can have two or more utterances, we construct all additional pairs where the source is the concatenation of the two utterances prior to the target utterance. We then merge the dataset with all these additional utterances. This results in another 50% increase in pairs over the merged MovieTriples-CMC dataset. The same process is applied to the validation dataset of MovieTriples.

B. Metrics

1) Perplexity: One well-established metric used to evaluate language models is perplexity. It explicitly measures the model's ability to account for the syntactic structure of each utterance [6]. The perplexity of an utterance $U = \langle w_1, ..., w_n \rangle$ is defined as:

$$Perplexity = 2^{-\frac{1}{n}\sum_{t=1}^{n}\log p(w_t|w_1,...,w_{t-1})}$$

In our experiments, we report the average perplexity over all validation sentences.

2) BLEU Score: Several recent works on dialogue systems adopt the BLEU score [14] for evaluation [15]. BLEU evaluates the quality of the predicted utterance by measuring the word overlap with the target utterance. We calculate the average BLEU score on the predicted utterances using the multi-bleu.perl script in Moses [16].

3) Human Judgments: Automatically evaluating the quality of dialogue response generation is still an open problem. Using perplexity may be unsuitable, since it does not evaluate the relatedness of the predicted utterance to the source utterance. BLEU score may also be unsuitable, since the range of the suitable responses is large, meaning that a suitable response may have a low BLEU score against the target utterance.

Therefore, we decide to evaluate the quality of the generated utterances in different models using human judgments, following the approach in [17]. Four labelers are given 100 random responses for each model along with the corresponding source utterances. They judge whether each response is appropriate and natural to the source utterance and assign them a label:

- **Suitable (+2):** the response is evidently an appropriate and natural response to the post
- **Neutral (+1):** the response can be a suitable response in a specific scenario
- **Unsuitable (0):** it is hard or impossible to find a scenario where response is suitable

We also evaluate agreement of the labelers by using Fleiss' kappa [18], as a statistical measure of inter-rater consistency.

C. Prediction Methods

1) Copying Mechanism: We replace every UNK token in the predicted utterance with the word in the source utterance it is best aligned with. The alignments are calculated using the attention mechanism, which produces alignment scores for every target token [19].

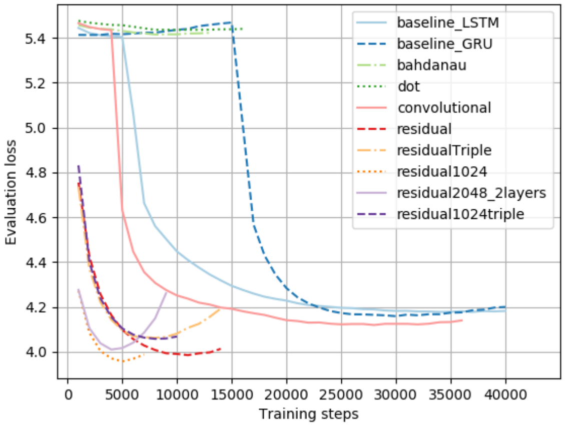

Figure 1. Average loss on the validation dataset during training for each model.

2) Beam Search Decoder: Instead of predicting the most probable token greedily, beam search keeps several hypotheses, or "beams", in memory and chooses the best one based on a scoring function. In addition, a length penalty factor is applied to the hypotheses, as described in [11], since without some form of length-normalization regular beam search will favor shorter results over longer ones on average. We keep seven beams at each step and we penalize with a length penalty factor of 1.0.

Model	Perplexity	BLEU
baseline_LSTM	4.83	0.73, 0.27
baseline_GRU	4.6	0.87, 0.15
bahdanau	8.12	0.03, 0.00
dot	6.96	0.00, 0.01
convolutional	4.92	0.98, 0.47
residual	5.38	0.96, 0.35
residualTriple	3.82	1.72, 1.39
residual1024	4.24	0.85, 0.50
residual2048_2layers	3.87	1.09, 0.40
residual1024Triple	7.63	0.99, 0.94

Table I
PERPLEXITY AND BLEU SCORES (COPYING MECHANISM, BEAM SEARCH)

D. Baseline Model

The baseline model uses a two-layered encoder and a four-layered decoder. The encoder projects the input sequence to a 512 dimensional vector. The hidden state of both the encoder and decoder is of size 512 as well.

As suggested in Google's sample configurations, we set the dropout keep probability on the input to 0.8 and to 1.0 on the output. We set the maximum length of source and target sequences to 80. In addition, the baseline model is trained on the original dataset only, that is, in the MovieTriples dataset.

As for the results obtained, Figure 1 shows, in comparison to the rest of the models, how the loss on the validation develops for `baseline_LSTM`. The model achieves a lower point of 4.178 on the loss after 35K steps, which takes 7 hours to train. This model obtains 0.73 on the BLEU score with the copying mechanism for prediction while it gets 0.27 with the Beam Search decoder. As far as perplexity is concerned, it gets an average perplexity of 4.83.

Observing this numbers, we imagine that the BLEU score and the perplexity are not very suitable metrics for this task. However, the poor results of the baseline also affect them. In the following sections, we show how we improve them a bit by improving the model even though these numbers still remain untrustworthy.

E. RNN Cell Variant

Even if the ”vanilla” implementation of the Sequence to Sequence paper uses LSTM cells, we have tried the very same configuration (see Section II-A) using GRU cells [20]. We believe that fewer parameters may speed up the training time.

When observing the best loss that each model reaches (see `baseline_GRU` in Figure 1), the GRU cells lead to a slight improvement with a value of 4.158 instead of the 4.178 of the baseline. However, the training time does make a difference as the model with GRU cells needs $5K$ steps less to converge, saving $1h$ of computation. The BLEU score improves to 0.87 on the predictions with copying mechanism but goes down to 0.15 with Beam Search. As for the perplexity, it's still quite similar, more precisely, 4.6.

Therefore, we keep building our improved model using GRU cells as it is computationally cheaper while maintaining the performance. Due to the problems we mentioned about BLEU and perplexity, we rely on the loss to decide which model is better.

F. Attention Mechanism

As mentioned beforehand, we have experimented with Badahnau and Dot attention mechanisms on the decoder. We keep the same configuration for the encoder. Then, we set the number of attention parameters to 512. Besides, we train these models with the merged dataset (see III-A2).

Also, for all the models with attention throughout this report, we do not pass the output of the encoder to the decoder anymore, we just initialize the first state of the decoder to zeros, instead.

In our experiments, both Bahdanau and Dot attention mechanisms show to perform quite poorly (see Figure 1, `bahdanau` and `dot`, respectively). The model using Bahdanau reaches a loss of 5.415 while Dot achieves 5.435. Note that this value is even worse than the initial loss of the baseline. Even though the former model needs less steps, 8K against 10K, each step is computationally more expensive and thus, the latter is $1h$ faster, needing 2.5 hours. The former model has perplexity 8.12 while the second has 6.96. When it comes to the BLEU scores, both obtain < 0.05 score with both prediction methods.

Due to the fact that the model using Bahdanau reaches a slightly better loss and the fact that it is suggested by Google in the sample configurations for large datasets, we go ahead building on this model.

G. Convolutional Encoder

Using convolutional encoder has proven to give fast and comparable results in the translation task [9]. Thus, we test this method in the dialogue generation task. It turns out to reach a lower loss value, 4.122, (check `convolutional` in Figure 1) compared to model using just Bahdanau attention (without the convolutional encoder). Moreover, the loss is slightly lower than the baseline model's. Contrary to what we expected, the model needs more training steps, 25K in particular, than the model using Bahdanau attention only and it takes 9 long hours. The predictions with copying mechanism give a 0.98 BLEU score whilst the Beam Search predictions get 0.47. The perplexity remains at 4.92.

H. Residual Connections

In this experiment, we compare the model with the Bahdanau attention mechanism only (see III-F) with a model that, apart from attention, also uses residual connections. We believe that they help to speed up the training time of deeper models which hopefully turns into better results.

This model (see $residual$ in Figure 1) reaches a loss of 3.984 which is a great improvement from the 5.415 loss of the simple model with Bahdanau attention. Moreover, it converges at 11K steps, in 3.5 hours, similar to the simple Bahdanau model. The BLEU scores obtained with the predictions are 0.96 with the copying mechanism and 0.35 with the Beam Search. Lastly, the average perplexity is 5.38.

Since residual connections have proved to be very appropriate for our task, we try to build on them in two different ways.

I. Three-turn structure

We trust that the three-turn structure may help the model learn better. Hence, we train the model with residual connections (see III-H) on the dataset preprocessed as explained in section III-A3. The behavior of the validation loss can be seen in Figure 1 under the name `residualTriple`.

The model reaches its lowest point on the loss at 4.061 after 8K steps and 4 hours. Its BLEU scores are 1.72 with the copying mechanism and 1.39 with the Beam Search. In addition, the average perplexity over all predictions is 3.82.

The three-turn structure does not seem to favor the loss but the perplexity and the BLEU scores are improved quite a lot. Still, as we only rely on the loss value, notice that the difference in the loss is very small.

Model	Mean Score	Suitable (+2)	Neutral (+1)	Unsuitable (0)	Agreement
baseline_GRU	0.755	40.5%	43.5%	16.0%	0.294
residual1024	0.728	39.8%	47.7%	12.5%	0.237

Table II
HUMAN JUDGMENTS RESULTS

Source sentence	Predicted by baseline model	Predicted by improved model
they want me to fly back tonight .	<person> .i ’ m not going to get out of the fly .	
nice .	i ’ m sorry .	<person> , you ’ re a good man .
yes . you got me there . that is mine .	i ’ m not going to be a UNK .	i ’ m not sure .
do it , mack . please . please . i ’ m begging you .	i ’ m sorry .	i ’ m not going to get out of this .
are you all right ?	you ’ re not a UNK .	i ’ m not .
you will have to prove it .	i ’ m not going to be a little man .	i ’ m not sure .

Table III
SAMPLE SENTENCES FOR BASELINE AND OUR IMPROVED MODEL

J. Deeper and Larger Models

We also want to analyze if a bigger and deeper model using residual connections turns into a better model. For the moment, we do not use the dataset with the three-turn structure.

In the first model, the encoder embeds the source sentence to a vector of size 1024. We increase the attention parameters and the hidden state of both the encoder and decoder to 1024 as well. Moreover, we increase the number of layers in the encoder to four. The performance of this model can be seen under the name residual1024 in Figure 1. It reaches a loss of 3.956 at step 5K after 2.5 hours of training which is the best we have got so far. However, the BLEU scores are not as good as in the model explained in III-I: 0.85 for the copying mechanism and 0.50 for the Beam Search. Yet, perplexity averages to 4.24.

Given the improvement on the loss that increasing the size of the model led to, we run another experiment increasing the embedding size, hidden state and number of attention parameters to 2048. However, this model needs a lot of space in memory so we decrease the layers on the encoder and the decoder to two. Again, we use the original structure of the dataset. It can be seen in Figure 1 (residual2048_2layer) that its performance is not that good, converging to a loss of 4.009 after 4K steps (2.5 hours). BLEU scores get to 1.09 with the copying mechanism predictions and to 0.40 with Beam Search. Regarding perplexity, it obtains 3.87. We believe that the performance gets worse because the model is not as deep as before. However, given our limited resources, we need to discard the idea of analyzing more.

As a final approach, we train the large model explained first in this section with the dataset exploiting the three-turn structure. However, as it can be seen in the Figure 1 (residual1024Triple), the performance is much worse. The model converges to a loss of 4.057 at 8K steps, after 3 hours. Its BLEU scores are better, though: 0.99 with the predictions using the copying mechanism and 0.94 with the Beam Search. The average perplexity goes up to 7.63, though.

It has already been mentioned the problems the perplexity and the BLEU score have. For this reason, we choose our final model based on the validation loss. Hence, we submit the first model explained in this section: the model with the basic encoder of four layers, Bahdanau attention and residual connections, decoder (whose state is initialized to zeros) with also four layers, 1024 embedding- and hidden-state size and 1024 attention parameters. The human judgment we developed on the predictions has also affected this decision, as we qualitatively think that this model gives the best predictions of all the models we have experimented with.

IV. QUALITATIVE EVALUATION

In order to qualitatively compare the baseline and the improved model, we choose only the model indicated as textttresidual1024 in Figure1. We observe that in the improved model, the size of distinct sentences is more than 3.9% of total sentences. But in the baseline model, the number is less than 1%. This means that our model comes up with more different sentences. Table II shows that both models are producing sentences of about the same quality according to the labelers, with fair agreement among them (0.21-0.40). Table III shows sample predicted sentences.

V. CONCLUSION

We tried to identify the main issues of the baseline model and tried several methods that claim to address them. According to our observations, the initial training time was high. To solve this, we tried using a convolutional encoder, GRU cells and dense residual connections which are elaborated in section III. Moreover, the predicted sentences were initially very general. ”i ’ m sorry”, ”<person> .” and ”i ’ m not going to be a UNK .” were among top 5 frequent sentences in baseline model. Our goal was to build a powerful model which can predict more meaningful sentences and achieve better scores according to our metrics III-B. We expanded our training data as explainded in III and used deeper and wider networks as reported in III-J to capture more complex connections.

References

[1] "tf-seq2seq," https://github.com/google/seq2seq. [Online]. Available: https://github.com/google/seq2seq

[2] D. Britz, A. Goldie, T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906*, 2017.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[5] C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs," in *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*. Association for Computational Linguistics, 2011, pp. 76–87.

[6] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[9] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[10] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, "Recurrent highway networks," *arXiv preprint arXiv:1607.03474*, 2016.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[12] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *arXiv preprint arXiv:1608.06993*, 2016.

[13] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[15] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," *arXiv preprint arXiv:1603.08023*, 2016.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[17] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[18] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[19] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," *arXiv preprint arXiv:1603.06393*, 2016.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.