# Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models

**Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville and Joelle Pineau**

**Problem and Motivation:**

Generally “End-To-End Dialogue systems” means single system that can be used to solve each of the four aspects, ie, natural language interpreter, dialogue state tracker, dialogue response selection, natural language generator, simultaneously. Typically, this system would take in a series of conversation(training set) and learns the interactions of the interlocutors to reproduce in later stage. Here a set of researchers try to implement one such model using an, Hierarchical recurrent encoder-decoder(HRED), introduced by Sordoni et al. HRED was originally introduced to assist in query suggestions. The authors examine thoroughly the implications of this model on a open domain conversational dialogue system using generative models. Generative models predict realistic, flexible interactions in a self reliant way. Further to analyse this model, they perform experiments with bidirectional HRED and also use bootstrapping technique, with various ways of word embeddings. Overall the goal was to show that HRED is competitive enough for this problem.

**Details of method:**

They characterize this problem as modelling a generative probabilistic model of utterances to unmask the interactive structure of the dialogue. HRED models the hierarchy of of sequences in a dialogue with two RNNs, one at the word level and one at the dialogue level.The RNN maps each utterances to an utterance vector. This dense vector is then mapped into a dialogue context, which is used to generate the tokens in the next dialogue. On the whole, the encoder RNN encodes the tokens appearing within the utterance and the context RNN encodes the temporal structure of the utterances appearing so far in the dialogue, allowing the information and gradients to flow over longer time spans. The decoder predicts one token at a time using RNN(Sordoni et al 2015). The comprehensive flow of the model is depicted in figure 1.

In HRED, utterance representation is in the last hidden state of encoder RNN. Since the dialogues contain more articulations compared to queries, the last state of encoder RNN may not reflect important information seen at the beginning of the utterance. Thus

they also try experiments on Bidirectional HRED, where both beginning utterance and last utterance are included to enumerate the context better.

Bootstrapping is used to get the commonsense knowledge in the dialogues of the two speakers. As the dataset is not sufficiently large. The model is initialized with parameters which was pretrained using a large corpus *SubTle.* Which contains large conversation dataset, where each interaction is question and answer form. They also introduce a dataset called as *MovieTriples.* It contains Movie scripts, which span a wide range of topics containing long interactions with few participants.

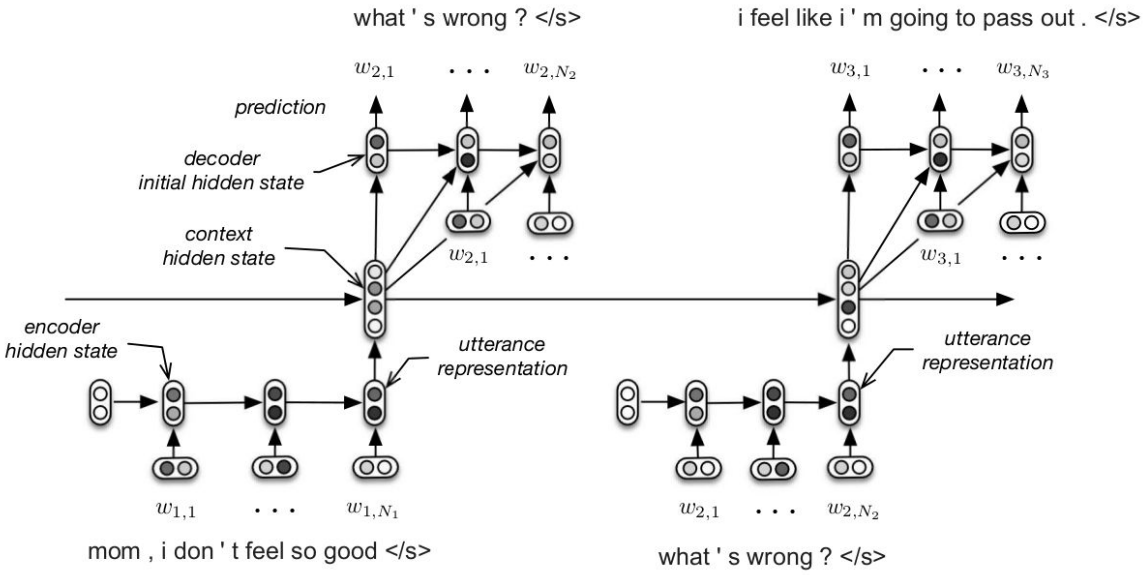


Figure 1: Computational graph of HRED with three turns of dialogue.

**Evaluation:**
Accurate evaluation of a open domain dialogue system is a challenge. The authors try to evaluate their model performance using **word perplexity**. It is defined by

$$\exp\left(-\frac{1}{N_W}\sum_{n=1}^{N}\log P_\theta(U_1^n, U_2^n, U_3^n)\right)$$

For a model with parameters $\theta$, dataset with N triples $\{U_1^n, U_2^n, U_3^n\}_{n=1}^N$ and $N_W$ the number of tokens in the entire dataset. Lower perplexity indicates a better model. As seen from the equation it is trying to find the syntactic structure of the dialogue.

**Results:**
Table 1 shows intensive results compared with various models and baseline. All the neural model outperform state-of-the-art n-grams. Without bootstrapping RNN models perform almost similar to DCGM-I and HRED models. The authors reason this due to size of the dataset. Where HRED and DCGM-I models are over fitting. The last 4 lines of the table show that bootstrapping the model shows a significant improvement in both measures. Overall, the bidirectional structure appears to capture and retain information from U1 and U2 utterances better than all the other models compared. The authors conclude by stating that, HRED has the potential to model long dialogues. They also conclude the use of a large external monologue corpus to initialize the word embeddings and the use of large related, but non-dialogue, corpus in order to pretrain the recurrent network increased the model performance.

| Model | Perplexity | Perplexity@$U_3$ | Error-Rate | Error-Rate@$U_3$ |
|---|---|---|---|---|
| Backoff N-Gram | 64.89 | 65.05 | - | - |
| Modified Kneser-Ney | 60.11 | 54.75 | - | - |
| Absolute Discounting N-Gram | 56.98 | 57.06 | - | - |
| Witten-Bell Discounting N-Gram | 53.30 | 53.34 | - | - |
| RNN | $35.63 \pm 0.16$ | $35.30 \pm 0.22$ | $66.34\% \pm 0.06$ | $66.32\% \pm 0.08$ |
| DCGM-I | $36.10 \pm 0.17$ | $36.14 \pm 0.26$ | $66.44\% \pm 0.06$ | $66.57\% \pm 0.10$ |
| HRED | $36.59 \pm 0.19$ | $36.26 \pm 0.29$ | $66.32\% \pm 0.06$ | $66.32\% \pm 0.11$ |
| HRED + Word2Vec | $33.95 \pm 0.16$ | $33.62 \pm 0.25$ | $66.06\% \pm 0.06$ | $66.05\% \pm 0.09$ |
| RNN + SubTle | $27.09 \pm 0.13$ | $26.67 \pm 0.19$ | $64.10\% \pm 0.06$ | $64.07\% \pm 0.10$ |
| HRED + SubTle | $27.14 \pm 0.12$ | $26.60 \pm 0.19$ | $64.10\% \pm 0.06$ | $64.03\% \pm 0.10$ |
| HRED-Bi. + SubTle | $\mathbf{26.81 \pm 0.11}$ | $\mathbf{26.31 \pm 0.19}$ | $\mathbf{63.93\% \pm 0.06}$ | $\mathbf{63.91\% \pm 0.09}$ |

Table1: Test results on three utterance dialogue

**Opinion and insights:**
To comment on the paper and the experimental analysis, the authors have done a splendid job. The paper is well written with extensive experiments giving intuition behind every experiment and reasoning out the results obtained.

As a take away points from the paper, I could mention 2 main points,(1) the use of hierarchy of RNNs using one to model sequence of utterances in the dialogue and one to model the sequences of tokens in an individual way. (2) the value of bootstrapping the model using **external data,** which make a significant difference to model performance. Bootstrapping uses 2 sources of data: general word knowledge and

domain knowledge. In order to learn the general understanding of the contextual meaning of words, the word embeddings are initialized using learned word embeddings from the Google News Dataset that have been trained using *word2vec*. A second level bootstrapping is to incorporate some general domain knowledge from a large(non-dialogue) corpus covering similar topics and type of interactions. They use the question-Answer *SubTle* corpus for this purpose.

There is one problem with the model, it produces generic responses such as “I don’t know” or “ I am sorry”. The authors reason it out saying, it may be due to data scarcity and these sentences are most appearing. I feel one possible way to eliminate this would be, remove these sentence on the whole or introduce a weightage for most frequently occuring sentences as done in TF-IDF. Also there should have been a different evaluation metric, as all the other metric compares the syntax more than context. I feel syntax is not equal to context always.