NLU Project 2: Beyond seq2seq Dialogue Systems

Florian Chlan,* Samuel Kessler,† Jovan Nikolic,‡ and Jovan Andonov§
ETH Zurich
(Dated: June 15, 2017)

This article explores the implementation and performance of a vanilla seq2seq model for the purposes of language modeling and machine dialogue. Furthermore, extensions are made to improve on the performance of this baseline model. In particular: the dataset is extended to include the Cornell Movie-Dialog Corpus, words are embedded using word2vec, the seq2seq architecture is adapted to encapsulate global attention and additionally with an anti-language model to encourage diversity in the responses.

I. INTRODUCTION

Deep learning has been very successful in tasks related to language modeling, machine translation, speech recognition and dialogue, with researchers edging closer to passing Turing's test. Today conversational agents mainly take the form of chat bots; used for a wide variety of tasks from ordering pizza to helping users cope with depression [1].

The following report outlines several approaches to machine dialogue. The baseline seq2seq model (see II B for technical background) produced poor results; repeated generic answers for different input sentences such as *"I don't know ."*, (see section IV for results). Subsequently, global attention (see II C for technical background) was utilized to address the seq2seq model shortcomings. The global attention model was able to produce more sophisticated replies, however still suffered from generic replies, (see section IV for further details). To produce a more diverse set of responses an element of randomness was embedded into the dialogue machine (see II D for technical details and IV for results).

II. TECHNICAL BACKGROUND

A. Recurrent Neural Networks

The recurrent neural network (RNN) is designed for learning sequences. In the dialogue context, an RNN recursively processes a sequence of embedded tokens (words or characters) $(w_1 \cdots w_T) \in V$, where V is the vocabulary and T is the input sequence length. At each time step the RNN updates its internal hidden state: $h_t = f(h_{t-1}, w_t) \in \mathbb{R}^n$, n is chosen a priori. The function f is usually a function of the form of an LSTM or GRU cell. The hidden state h_t encapsulates all prior information regarding previous tokens. Hence, the hidden state can be projected via a softmax function onto the space of the vocabulary, of size $|V|$ for purposes of prediction. Crucially, the number of iterations where the network is unrolled needs to be known a priori: this presents a severe limitation for dialogue models.

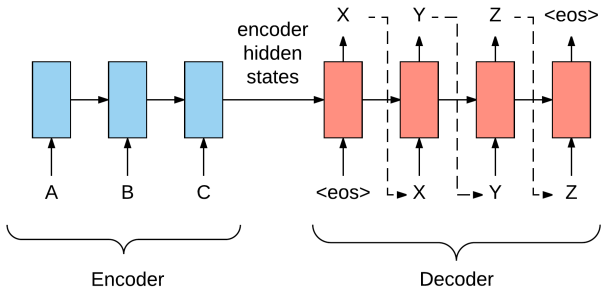

FIG. 1. This is a sequence to sequence model for a conversational or translation model. <eos> marks the end of a sentence.

B. Sequence to Sequence Models

Sequence to sequence (seq2seq) models learn how to map variable length sequences to sequences. Thus their ability to support conversational and translation models. The seq2seq model employs a first RNN network, the encoder, to map an input sequence to a fixed length vector $v \in \mathbb{R}^n$ and a second RNN network, the decoder, to map the vector to the target sequence. For additional details see [2].

C. Global Attention

For seq2seq models the encoder embeds the input sequence to a fixed length vector. This embedding presents a bottle neck. The performance of the decoder and the encoder degrades for longer sentence lengths.

Global attention aims to rectify this bottleneck by allowing the decoder to condition on all encoder hidden states. Intuitively, the decoder may choose certain parts of the encoded source sentence to focus on, when out-

* fchlan@student.ethz.ch
† sakessle@student.ethz.ch
‡ jovan.nikolic@gess.coss.ethz.ch
§ andonovj@student.ethz.ch

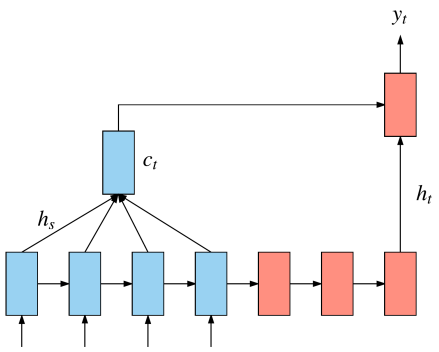

FIG. 2. Global attention model; a context vector, c_t, is calculated as a weighted average of source hidden states. The context vector is conditioned upon, on prediction of y_t by the decoder. h_s are the hidden states of the source sentences i.e. those of the encoder. h_t are target hidden states i.e. those of the decoder.

putting a reply.

The RNN graphical model defines a probability distribution $p(y)$ over the targets, which factorizes at every time step. Crucially, each conditional probability is conditioned on the context vector $c = f(\{h_1, \cdots h_T\}) \in \mathbb{R}^n$, where f is a weighted sum.

$$p(y) = \prod_{t=1}^{T} p(y_t|\{y_1, \cdots y_{t-1}\}, c) \tag{1}$$

For further details regarding global attention see [3].

D. Anti language model

To encourage more diverse responses for a dialogue machine a maximum mutual information is used as an objective function for the neural network, see [9] for details. Targets are generated which maximize:

$$P(T|S) - \lambda P(T) \tag{2}$$

where S is the source sentence, T is the target sentence, $P(T|S)$ is the probability generated from the seq2seq decoder, $P(T)$ is the true distribution of targets, λ is a penalization hyperparameter which only penalizes predictions up to $t = k$.

III. TRAINING PROCEDURES

The implementation was performed in TensorFlow. An existing TensorFlow implementation [4] formed the basis of the realization of the models introduced in this section. Cross entropy was used as the loss function. Adam optimization was employed in conjunction with the truncation of gradients larger than 10. Dropout with a probability of 0.5 was also introduced into the network to prevent overfitting.

A. Experiment Baseline

a. Training data For all experiments a baseline training dataset from the Movie-DiC corpus was used. The corpus comprised over 392k conversation tuples. For sentences larger than 60 words long, the entire conversation tuple was removed from the training dataset. For each tuple, decoder targets were prepended with beginning-of-sentence tag, <bos>. Decoder targets were appended with the end-of-sentence tag, <eos>. Training was performed for 20 epochs.

b. Network variant experiments Experiments on the baseline were performed using a 1, 3 and 5 stacks of RNNs for the encoder and decoder which showed little improvement in the output. Additionally a bidirectional encoder was used, like in [2], with little improvement in the outputs. However, the paper was demonstrating a translation model and not a dialogue model, the author's discussion on the effectiveness of the bidirectional encoder is only relevant to a translation model. Network variant experiments were run for over 20 epochs with 20% of the training data.

c. Implementation A vocabulary size of 10k was used out of over 67k unique words. Words were embedded into vectors in $\mathbb{R}^{200}$ using TensorFlow's "embedding_matrix" via a simple partitioning of words into the embedding space. Single unidirectional RNNs were employed by the encoder and decoder. LSTM cells with a hidden state of size 512 were used as a basis of both encoder and decoder RNNs.

B. Global Attention

a. Cornell Movie-Dialog Corpus The Cornell corpus was pre-processed to mimic the Movie-DiC dataset. The Cornell dataset comprised over 221k new sentences which could be added to the training dataset. Spaces between words and punctuation were added. All numbers were substituted with the <number> tag. All names were replaced with a <person> tag.

b. Word2vec The word2vec algorithm [6] was used for embedding words to vectors of size 200. word2vec was implemented with the Continuous Bag of Words (CBOW) algorithm, with a negative sampling of size 5. A vocabulary size of 10k was used. Words with a minimum frequency of 1 were embedded for 5 epochs. The embedding was trained on the Movie-DiC and Cornell corpses.

c. Implementation Global attention as described in [3] was implemented in TensorFlow version 1.0. Training was performed on 20 epochs.

d. Genres For each sentence in the Movie-DiC and Cornell data set, the genre of the original movie was prepended, as an additional feature for the network to learn from. Different genres will influence the dialogue of a movie.

C. Anti Language Model

a. Implementation An existing implementation [8] inspired the anti language model. The Cornell data set, word2vec and global attention extensions were utilized. Inputs were fed into the encoder in reverse order, in line with [8]. A 3 layer encoder and decoder was used. The model was implemented in TensorFlow version 1.1.

b. Training Scheduled sampling is used for training: with a probability of 0.75 the decoder feeds in the ground truth at time step t and with a probability 0.25 it feeds in the softmax argmax from $t-1$ [7]. This is to improve the robustness of the predictions. Training was performed on 10 epochs.

c. Penalized Predictions For one particular conversation tuple, $P(T)$ in equation 2, is estimated by passing as many <pad> tokens into the encoder as there are input words [8]. Accordingly, $P(T)$ was estimated for each encoder input length. This probability can then be subtracted from the regular output probabilities over our vocabulary to penalize generic predictions and boost less generic predictions. The penalization is performed greedily at every time step. To add an extra degree of randomness the first token is sampled using a multinomial distribution, with the penalized logits as the underlying probability mass function. Doing this sampling for further than the first time step destroyed grammar in the predicted sentences. The hyperparameters λ and k were chosen manually, final values were 0.1 and k=4 i.e. the penalization was applied to the first 4 predicted tokens only. Higher values of λ were also observed to destroy the grammar in the predicted sentences.

IV. RESULTS

a. Prediction For prediction in the network, the encoder is fed a source sentence which is subsequently encoded to a fixed length vector $\mathbb{R}^{200}$ and passed to the decoder in addition to a single <bos> tag at $t = 1$ to initiate the prediction process in the decoder RNN.

b. Model measurements The median perplexities have been calculated across the entire evaluation data sets for each model, table I. The evaluation dataset is greater than 49k sentence pairs. Also, selected predictions have been chosen to demonstrate model performance, table II.

The perplexity is calculated as follows:

$$Ppl = 2^{-\frac{1}{T}\sum_{t=1}^{T} log_2 p(w_t|w_1 \cdots w_{t-1})} \quad (3)$$

where T is the length of the target sentence. The decoder is left to unroll until it reaches the <eos> token, and $p(w_t|w_1 \cdots w_{t-1})$ is the probability of the ground truth token w_t looked up from the softmax distribution output by the decoder at time t. Intuitively, the perplexity can be seen as a measure for how different a predicted sentence is to the evaluation set. The smaller the perplexity the greater the fidelity to the evaluation set.

A. Discussion

Regarding perplexities, shown in table I, the extensions made are seen to decrease the perplexities of our evaluation data set: the language model extensions are seen to improve learning. Significant improvements were observed for the anti language model predictions. However the reported perplexities are an order of magnitude greater than perplexities presented in recent conference papers [9] [11].

The observed predictions in table II are more sobering; a significant proportion of the baseline predictions are very generic e.g. <person> . <eos>. The attention model extensions provide slightly more variety in the replies, nonetheless far from the mark, in comparison to the ground truth. Amusingly, the anti language model provides more diversity in the vocabulary of the reply, at the cost of a meaningful and grammatically correct replies.

Model	Ppl
Baseline	373
Baseline + Cornell + word2vec + global attention	327
Baseline + Cornell + word2vec + global attention + genres	337
Baseline + Cornell + word2vec + global attention + anti language model	271

TABLE I. Results: median perplexities across the validation data set for stated models.

In general, the results are poor for the sophistication of the system. The perplexity values are high and a significant proportion of the predicted sentences from the decoder are very generic. Despite reusing working code from [4] and [8] it is accepted that there might be a bug tarnishing the experiments described above.

B. Improvements

A larger corpus and longer training times would improve results, state of the art models are based on the order of $10^6 - 10^7$ sentence pairs [10] [9]. The potential bug needs to be addressed. Additionally incorporating reinforcement learning [12] and in particular, beam-search would lead to improved results [13].

Context	Response
goodbye , young fellow . we ' ll see what we can do .	GT: i ' m not worried . everything will be fine . the old team together . nothing can stop us . she turns and walks out of the shot . ¡person¿ stands for a second watching her , then turns to his assistant . B: <person> . <eos> BA: bathroom . <eos> BAG: i ' m not going to see you again . <eos> BA+antilm: going earth ve sing ? <eos>
me either . i wish i could . . but . want me to shut these , sun comes in pretty strong in the morning .	GT: sure . . . thanks . . . B: <person> . <eos> BA: <person> . <eos> BAG: you ' re right . <eos> BA+antilm: yourselves best , saturday , i ' retreat a <unk> . <eos>
well , i don ' t know anyone named obi- wan , but old ben lives out beyond the dune sea . he ' s kind of a strange old hermit . <continued_utterance> i wonder who she is . it sounds like she ' s in trouble . i ' d better play back the whole thing .	GT: he says the restraining bolt has short circuited his recording system . he suggests that if you remove the bolt , he might be able to play back the entire recording . B: <person¿ . <eos> BA: <person> . <eos> BAG: you ' re a very attractive woman . <eos> BA+antilm: fame <person> , <person> . <eos>
what ' ve you done ?	GT: nothing . B: i ' m going to get a job . <eos> BA: i ' m not sure . <eos> BAG: i ' m not sure . <eos> BA+antilm: away and notice about the <unk> . <eos>
nobody . it used to be that sex was the only free thing in the world . no longer . ¡person¿ , alimony . all financial . ¡person¿ ' s an illusion .	GT: it ' s the only illusion that counts . B: <person> . <eos> BA: <person> . <eos> BAG: you ' re a monster . <eos> BA+antilm: dude more problems there . <eos>
sometimes i get boners so big i can ' t hide them . and then i get nervous and my heart starts pounding , and it all just , like , feeds my boner . it just becomes this . thing . that ' s attached to me . and it won ' t go away .	GT: just imagine if girls weren ' t <unk> out by our <unk> and they actually wanted to see 'em , like this shit . B: <person> , i ' m sorry . <eos> BA: i ' ll kill you . <eos> BAG: you ' re not going to kill me . <eos> BA+antilm: not interested sing throws . <eos>

TABLE II. Dialogues. GT is the ground truth, B is the baseline, BA is the baseline model + word2vec + Cornell + global attention. BAG is BA + genres. BA+antilm: BA + anti language model.

[1] Stanfords Woebot is a therapy chatbot for depression and anxiety - Business Insider. (n.d.). Retrieved June 9, 2017, from http://uk.businessinsider.com/stanford-therapy-chatbot-depression-anxiety-woebot-2017-6?r=US%5C&IR=T

[2] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Retrieved from http://arxiv.org/abs/1409.3215.

[3] Bahdanau, D., Cho, K., & Bengio, Y. (n.d.). NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE. Retrieved from https://arxiv.org/pdf/1409.0473.pdf

[4] Matvey Ezhov https://github.com/ematvey/tensorflow-seq2seq-tutorials

[5] Cornell Movie - Dialog Corpus https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (n.d.). Distributed Representations of Words and Phrases and their Compositionality. Retrieved from http://web2.cs.columbia.edu/~blei/seminar/2016_discrete_data/readings/MikolovSutskeverChenCorradoDean2013.pdf

[7] Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (n.d.). Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. Retrieved from https://papers.nips.cc/paper/5956-scheduled-sampling-for-sequence-prediction-with-recurrent-neural-networks.pdf

[8] Marsan-ma https://github.com/Marsan-Ma/tf_chatbot_seq2seq_antilm

[9] Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016). A Diversity-Promoting Objective Function for Neural Conversation Models. Retrieved from https://arxiv.org/pdf/1510.03055.pdf

[10] Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. Retrieved from https://arxiv.org/pdf/1508.04025.pdf

[11] Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., & Dolan, B. (n.d.). A Persona-Based Neural Conversation Model. Retrieved from https://arxiv.org/pdf/1603.06155.pdf

[12] Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., & Jurafsky, D. (n.d.). Deep Reinforcement Learning for Dialogue Generation. Retrieved from https://arxiv.org/pdf/1606.01541.pdf

[13] Wiseman, S., & Rush, A. M. (n.d.). Sequence-to-Sequence Learning as Beam-Search Optimization. Retrieved from https://arxiv.org/pdf/1606.02960.pdf