CWord: Incorporating Larger Context in Neural Conversation Model

Felipe N. Ducau
Center for Data Science
New York University
fnd212@nyu.edu

María Elena Villalobos Ponte
Center for Data Science
New York University
mvp291@nyu.edu

Abstract

We explored the task of building a context-aware, open domain, non-goal driven conversational agent. This problem is non-trivial due to data sparsity and remains one important open question in the field. For this Natural Language Generation task in particular, there is a vast diversity of valid responses to a source utterance, so conditioning on larger context should help produce more on-topic and semantically relevant responses. In support of this goal, we propose an Attention based Encoder-Decoder that depends not only on the previous utterance, but also in the conversational context. Our proposed model performs similar to non-context aware architectures in terms of automatic evaluation, but generates context-aware more human like responses when evaluated by a human.

1 Introduction

Recently, data-driven dialog systems have become a prominent research area. There are many applications for conversational agents: helping the user to complete a specific task, such as buying movie tickets or writing automated answers for emails. Commercial solutions are often highly domain-specific and constrained to a very particular task, relying mostly on a set of predetermined rules. However, recent development of deep neural networks has made it possible to propose alternative solutions with interesting results.

Non-goal driven dialog systems can be interesting for multiple applications like learning a foreign language or playing the role of characters in videogames. It has also been proposed to use then in training goal-driven conversational agents by playing the human role.

One of the many remaining open questions related with conversational systems is how to incorporate contextual knowledge from previous stages of a conversation in order to generate more relevant responses. It is difficult to imagine that a system could maintain a fluent conversation with a human without taking into account this information. The topic of a conversation is naturally embedded in its context, which allows to reduce ambiguity from words and keeps track of factual information.

In this work we propose a novel architecture which aims to incorporate contextual information with small computational overhead oriented for (but not limited to) non-goal driven conversational agents. For this, we use a generative approach to model spoken dialogs with Neural Networks.

We begin this work by introducing some necessary definitions, relevant technical background and current challenges of the field in section 2. In section 3 we explore the available corpora for training conversational models and provide some general guidelines to take into account when choosing a training dataset.

Section 4 provides a survey of the most relevant works in the field followed by a description of an alternative approach to incorporate conversational context to current neural network architectures in a computationally efficient manner. In section 6 we evaluate this model both in a quantitative and qualitative way and show some sample utterances generated by the model. Finally we finish this work by

drawing some conclusions about this work in particular and the field in general.

2 Technical Background

2.1 Retrieval-based vs. Generative models

Retrieval-based models use a repository of possible responses and select deterministically which is the best one given an input statement or query. They do not generate new text and will never generate an answer outside this fixed set of possible responses. Generative models, on the other hand, do not rely on predefined answers, they generate it from scratch using the posterior probability of possible utterances. This can be seen as a probabilistic language model and is often associated with the neural translation task.

2.2 Goal-driven vs. Non-goal driven

Goal-driven conversational agents have a specific task to perform, a typical example would be a an IT helpdesk troubleshooting agent that interacts with the user to help her solve a prticular problem. This task is related with a closed domain in which the knowledge to take into account to generate the answer has a very limited scope. A non-goal driven agent, on the other hand, does not have a specific task to attain other than establishing a human-like conversation. This task is often associated with an open domain, in which the human counterpart can take the conversation potentially everywhere.

2.3 Sequence to Sequence models

The sequence-to-sequence structure (SEQ2SEQ) is a general end-to-end approach to sequence learning (Sutskever et al., 2014). In these models, the input sequence is mapped into a vector of fixed dimensionality with a Recurrent Neural Network (RNN). After that, a second RNN is used to decode the target sequence from the vector representation of the encoded sequence. The strength of this relies in its simplicity and generality, given that no assumptions are made about the sequence structure.

Since it is well known that plain RNNs suffer from several problems, the standard is to use Gated Recurrent Units (GRU) (Cho et al., 2014a) or Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) recurrent neural networks.

SEQ2SEQ architectures are often referred to as RNN Encoder-Decoder (Cho et al., 2014b). The encoder is an RNN that reads an input sequence and builds a vector representation which is later fed to the decoder (a second RNN) which generates an output sequence. Notice that the decoder part of this model resembles a Neural Language Model (NLM) (Mikolov et al., 2010), but instead of only taking as an input the previous words of the sentence, the output is also conditioned on the encoded input sequence.

Encoder-Decoder structure was introduced with the problem of machine translation in mind. In this case the input sequence or *source sentence* is in one language while sequence generated by the decoder or *target sentence*, is in different one.

2.4 Attention Mechanism

In the attention mechanism (Bahdanau et al., 2014) a bidirectional RNN (Schuster and Paliwal, 1997) is used in the encoder. The produced vector at each position in the sequence can be interpreted as an *in context* encoding of the word in that position, since it contains information from both its preceding and following words. In the decoding stage, the decoder chooses to which parts of the source sentence to pay attention to. This alleviates the encoding task of building a single representation of the source sentence into a fixed-length vector. In the case of translation this helps with the word alignment between the source and target sentences.

2.5 Evaluation Metrics

Finding a well-suited objective function for validation in neural conversation models remains an open question in the field and the current standard is to use the minus log likelihood of the data under the model. As in NLM, perplexity is usually used as one of the primary metrics. The reason being that the decoder should be able to generate grammatically correct sentences.

Other widely used evaluation metrics at test time include BLEU (Papineni et al., 2002) and METEOR scores (Banerjee and Lavie, 2005), both *borrowed* from the Neural Machine Translation (NMT) field. The problem with these metrics is that they satisfy only one direction of the Schatzmanns criteria: a high BLEU or METEOR score implies that the

model is generating solid outputs, but the converse is not true.

In order to evaluate how aligned a generated response is with a source utterance, similarity metrics based on word embeddings were recently proposed and nicely summarized in Serban et al. (2016a). The *Embedding Average* metric projects both the generated utterance and the ground truth into two separate real-valued vectors by taking the mean over their word embeddings, and then computes the cosine similarity between them. The *Embedding Extrema* metric is similar in spirit, but embeds the responses by taking the extremum of each dimension, and then computing the cosine similarity between them. Lastly the *Embedding Greedy* uses cosine similarity between word embeddings to find the closest word in the ground truth sentence for each word in the model generated response. Then, given this alignment, it computes the mean over the cosine similarities for each pair of words. Even though these metrics are not strongly correlated with human judgements, they can be interpreted as meassuring the topic similarity. For reproducibility purposes this is computed using Word2Vec word embeddings Mikolov et al. (2013).

3 Data

Available corpora for data-driven conversation models training is extensively documented in Serban et al. (2015a). The different aspects to take into account when choosing a dataset are also discussed in detail, in this section we discuss some of those aspects briefly.

3.1 Spoken vs. Written Language

Dialog systems can interact with the user in multiple ways, they can have an integrated text-to-speech synthesizer that products spoken language or they can also interact with the user by directly using text, in which case the communication would be written. In general, language varies with respect to has communication happens, spoken language tends to be less formal, has lower information content and contains many more pronouns than written language. For this reason, depending of the type of interaction that the system will have: written, spoken or multimodal, different datasets will be better suited for training. Notice that this distinction only applies to the original source of language, if a transcript is made of originally spoken language it will preserve the properties of spoken language. Similarly, if the final system will communicate with the user by spoken language, the written output should preserve the characteristics of this communication mode.

3.2 Human-Human vs. Human-System Corpora

Datasets are also distinguished by the nature of interactive parties. In some cases the human interacts with a computer and in other cases the human interacts with another human. Since current systems are significantly constrained, these two types of conversations are expected to be very different. It seems to still be debatable if goal-driven systems should be trained with human-human or human-systems corpora, but for non-goal driven and open domain dialogs clearly human-human dialog corpora is preferable because they reflect natural dialog characteristics. Conversations should not be constrained to a particular subject and participants are not asked to carry out specific task.

3.3 Corpora from Fiction

Works of fiction like novels, movie scripts and subtitles are a vast source of dialog data, but these are artificial sources of data and training data should be as natural as possible. Surprisingly, studies have shown that these artificial dialogs like spoken language in movies resemble human spoken natural language. This could be due to the fact that these dialogues are, by design, intended to sound like natural spoken conversations. One possible issue with these data is that the conversation depends upon events that occur outside of the spoken language, for example in movies the visual component can present cues on how the conversation should be driven and by using only transcripts, we are restricted to what is being said.

Several datasets, such as the Movie DiC Corpus (Banchs, 2012) containing about 130,000 dialogues from movie scripts extracted from the Internet Movie Script Data Collection and the Movie Triplets Dataset (Serban et al., 2015b) which is a derivation of the former are the main candidates. Other interesting datasets to evaluate are the Cornell

Movie-Dialogue Corpus (Danescu-Niculescu-Mizil and Lee, 2011) which has short conversations extracted from movie scripts, and the Filtered Movie Script Corpus (Nio et al., 2014) that consists mostly of raw scripts.

3.4 OpenSubtitles Dataset

The OpenSubtitles parallel corpus (Lison and Tiedemann, 2016) is a collection of subtitles from movies in 65 languages in XML format. It corresponds to the Corpora from Fiction introduced earlier, it is also a transcript of spoken language between human participants. Movies of multiple genres and topics are used, so we expect characters to discuss several subjects and ideas, making it very appropriate for our particular task. Only English movie subtitles were relevant for our purposes which correspond to 337M sentences.

This dataset is also used by Sutskever et al. (2014) in the Neural Conversational Model, which is to the best of to the best of our knowledge and opinion the model that has shown the best qualitative results. Notice that, since multipla utterances correspond to a single movie the data split into training, validation and test sets has to be done carefully in order to assign all utterances from the same movie to the same dataset split. In order to make training time feasable we used only a small fraction of the original dataset corresponding to 2M utterances.

3.5 Ubuntu Dialogue Corpus

The Ubuntu Dialogue Corpus (Lowe et al., 2015) is a dataset containing 930,000 dialogs composed by 7,100,000 utterances and 100,000,000 words. It was extracted from Ubuntu chat logs, so it is restricted to two participants which are both human. The dataset has both the multi-turn property and unstructured nature of interactions. Even though this dataset is better suited for a goal-driven task, like helpdesk troubleshooting, it was used in this work to be able to compare model results in a quantitative way.

4 Related Work

In this work we approach the problem of response generation as statistical machine translation problem as first framed by Alan Ritter (2011). As an analogy to how Encoder-Decoder RNN models have been used in Neural Machine Translation (NMT), they were proposed as a data-driven, unsupervised approach to building conversational models which can be trained end-to-end. Recently, work by Sordoni et al. (2015b) and Shang et al. (2015) used RNNs to model dialogue in short conversations. Later Vinyals and Le (2015) applied a direct end-to-end SEQ2SEQ model with LSTM neural networks trained with a parallel corpus of 62M of utterances and a vocabulary size of 100K words obtaining promising results given the simplicity of the model.

In order to be able to generate an accurate response to a given utterance in a conversation, it is intuitively important to take into account the topic and content of the conversation so far to reduce the search space of valid candidate response utterances. This problem was studied for the case of Language Models, where structures to make use of larger context like Late Fusion (Wang and Cho, 2015), which adds a context vector into a GRU cell in a particular way and Contextual LSTM (Ghosh et al., 2016) have been introduced, both showing interesting results in the tasks they define.

Closer to our line of work, and probably the natural extension to RNNs to take context into account the Hierarchical Recurrent Encoder-Decoder (HRED) (Sordoni et al., 2015a) was proposed for context-aware search queries generation. HRED consists of three RNNs: an *encoder* RNN, a *context* RNN and a *decoder* RNN, where the first one generates a real valued vector which is fed as an input to the *context* RNN that produces a vector representation of the context. Finally the output sequence is generated by the *decoder* RNN by conditioning on the state of the context vector representation. This model is also very expensive to train and needs to make use of truncated backpropagation through as an approximation to actual complete backpropagation. Even thought this approach to introduce the context fails (at least in terms of its scoring function), it shows that the architecture benefits highly from bootstrapping the learning from a larger corpus and from pretrained word embeddings, achieving increases of almost 10% in terms of perplexity.

This same model was then applied to the task of non-goal driven utterance generation (Serban et al., 2015b), achieving slightly worse results than a vanilla RNN in terms of perplexity. Based in this

architecture the Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED) (Serban et al., 2016b) was proposed. This architecture makes use of a high-dimensional stochastic latent variable at each dialogue turn. The main idea of this latent variable is to model ambiguity and uncertainty.

5 Context-Word Concatenation Model

In this section we present a new approach to use context information for neural conversation models, in particular for the task of response generation. Unlike previous approaches, which use the context close to the generation stage of the model, we add a vector representation of the context in the encoder part of an Encoder-Decoder architecture. The idea behind this approach is that a neural network model would benefit the most from the context if it helps it understand the input words better, and then produce a response, instead of using the context representation as a space in which project the already decoded output.

For instance, in a conversation about operating systems, if the encoder "sees" the word *windows* and the context at the same time, it might be able to understand that it refers to a company instead of an opening in the wall. On the contrary, if the encoder already encoded the sentence into a vector, that difference might be much harder to capture and the gain of using context might be reduced.

This interaction between the context and the input words is also aligned with how RNNs work. In GRU for exaple, in order to compute both the *reset gate* and the *update gate*, for all practical considerations, the context vector h appears in concatenation with the word embedding of the input word.

Early experiments shown that this kind of models benefits from the incorporation of the attention mechanism applied over the source utterance. We believe that the use of longer context will be achieved more effectively if the decoder is allowed to adaptively choose a subset of vector representations produced in the encoding.

Figure 1 shows a simplified diagram of the proposed model in which we use the last n utterances of a conversation as the context to generate the response utterance u_t. As a first step we take the mean of the embedding vectors for the words of the context utterances u_{t-n-1} to u_{t-2} to create the context vector c_t. Then, we encode the sentence u_{t-1} with a bidirectional GRU structure, in which we concatenate each source of word $w_{t-1}^{(i)}$ with the context vector c_t. Once we have the outputs of the bidirectional encoder we generate our response utterance with a decoder RNN which is initialized with the concatenation of the to end hidden states of the encoder $[\overrightarrow{h}_{t,N_{t-1}}; \overleftarrow{h}_{t,N_{t-1}}]$ and doing attention over the concatenation of the hidden states of the bidirectional encoder. We will refer to this model as *CWord* for short.

It is worth noticing that the embedding matrix does not need to be necessarily the same for the context and the input utterance and could be potentially useful to use a different embedding matrix for this purpose. Since it is one of our goals not to increase the number of parameters of the model except when extremely necessary we have not implemented this option but it is one of the natural extensions for future experiments.

6 Experimental Evaluation

6.1 Experimental Setup

For the evaluation of the model we consider the task of generating utterances using as side information what has been said in the previous four turns of a conversation in an open domain setting. We compare the results of our model with a GRU Encoder-Decoder architecture with attention which shares the exact same architecture as our Context-Word Concatenation Model except for the changes introduced in the previous section.

Both the GRU units used and the attention mechanism follow the implementation suggested in Cho et al. (2014a) and in Bahdanau et al. (2014) respectively.

We consider two different training sets. The first one is a subset of the OpenSubtitles (Lison and Tiedemann, 2016) which is well suited for the problem at hand as explained in section 3. Given the lack of a standard dataset for open conversation systems, we also trained the two models in the Ubuntu Dialogue Corpus Dataset (Lowe et al., 2015) for comparison purposes.

We trained both models in the Ubuntu Dataset with with little preprocessing: first we added the end

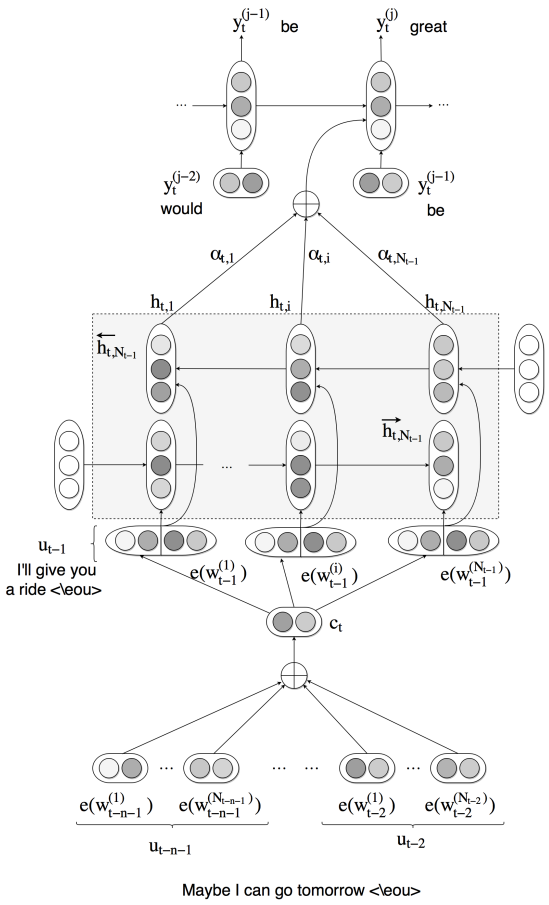

Figure 1: Computational Graph of CWord Architecture

of turn token (*__eot__*) at the end of each line, and the end of dialog token (*__eod__*) at the end of each file. After that, the tokenization was done at word level splitting by blank characters and punctuation symbols.

We also used two sub-datasets of the OpenSubtitles Dataset, one consisting on 2M (OS2M) of utterances while the second one containing 5M (OS5M). The raw data available at OpenSubtitles website is in XML format. We parsed those files with simple rules to determine the change of turn in the conversations and the end of each conversation, which was created by inspection and does not prentend to be extensive: (1) every time there are more than 60 seconds of silence we consider that the conversation has ended and introduce the end of dialog symbol, (2) each time that at the begining of an XML time tag we see an 'S' we decide that a new turn has started and insert the end of utterance symbol before processing the text, and (3) every time we see an 'E' at the end of a XML tag we consider that the turn has ended and insert a end of turn symbol. Furthermore, since there are several subtitles for the same movies we only extract information for one of the files corresponding to that movie and we also make sure that when we do the splitting into train, validation and test sets, dialogs from the same movie can not appear more than one set.

The splitting of all the datasets was done by assigning 80% of the obsevations for the training set and 10% for the validation and set and the train set each.

All the models were trained using word embedding dimension of 620. The size of the GRU units is used was 1,000. We considerred a vocabulary size of 20,000 words and a batch size of 128.The whole model was trained using backpropagation and optimized with Adadelta (Zeiler, 2012) applying the early stopping criterion based on the validation set perplexity with a patience of 5.

For the models that use context we conditioned the output on the previous 4 utterances. In total we trained 5 models,

- **OS_noCTX_2M**: using OS2M dataset without considering context.
- **OS_CWord_2M**: using OS2M dataset and CWord model.
- **OS_CWord_5M**: using OS5M dataset and Cword model.
- **UB_noCTX**: trained with Ubuntu Dataset without considering context.
- **UB_CWord**: CWord model trained with Ubuntu Dataset

The results for the experiments in the OpenSubtitles dataset are summarized in table 1.

6.2 Quantitative Evaluation

OpenSubtitles Datast

We observe that our proposed architecture performs really similar than the plain Encoder-Decoder architecture in terms of all the computed metrics, even though slightly worse in most of them. We also note that increasing the amount of data, boosts the

performance of the CWord model in a significant way which leads us to think that the performance of both CWord and Encoder-Decoder architectures reached its maximum for the given dataset size. This behavior of continued reduction of perplexity with dataset size can be clearly appreciated in Vinyals and Le (2015), where training with a deep Encoder-Decoder without attention reaches a perplexity of 17 in the entire OpenSubtitles Dataset.

Ubuntu Dataset

Table 2 shows the results obtained for the Ubuntu Dataset and compares against the 3-turn (2 context) implementation in **?**. We Again see that using a plain Encoder-Decoder network achieves the best overall resutls in the considered metrics. One thing to note is the difference in the Embedding Extrema metric between with our implementation reaches much better results, but overall there is not a clear best model.

The results (table 1) show that

6.3 QualitativeEvaluation

OpenSubtitles Dataset

We also test our models in an interactive way in order to appreciate if human-like conversations were being produced and how useful was the conversational context in giving on-topic, semantically relevant responses in a dialog. Below we show a sample conversation using 5M utterances of the Opensubtitles dataset as training.

Dialog 1

Human: hello .
Machine: hello .
Human: hi !
Machine: hello , *UNK* .
Human: how are you doing ?
Machine: you 're looking good .
Human: thank you ! you don 't look so bad yourself .
Machine: i 'm fine .
Human: what where you doing this morning ?
Machine: i 'm looking for a new UNK .
Human: tell me more about it !
Machine: i 'm not going to tell you .

Dialog 2

Human: It is too cold outside .
Machine: I 'm going to get a coffee .
Human: maybe I 'll join you . I also want to have something to drink .
Machine: I want to drink .
Human: what do you want to drink ?
Machine: I want to drink .
Human: what ?
Machine: I 'm going to get a **drink** .
Human: what kind of drink ?
Machine: a UNK .

Dialog 3

Context: what is you favorite trick ? I love magic . It reminds me of my childhood .
Human: Do you believe in magic ?
Machine Top Alternative responses:

- I believe in magic .
- I don't believe in magic .
- Yeah, I believe in magic .
- Yes , I do .
- I believe in magic , **too .**

The sentences are well constructed, have a reasonably accurate grammatical structure and are fairly appropriate given the previous utterance. The produced utterances are usually different than the ones produced by the Atention-Based Encoder Decoder model. In particular, when the same utterance is given as an input more than once, the generated sentence is different. We picked these particular examples because we found some interesting details.

In **Dialog 2** given the context about drinking coffee, the model outputs a relevant utterance to a sentence that does not include information about this subject. This means that the response was necessarily produced by taking the conversational context into account. In **Dialog 3**, when reviewing the top probable responses we found a utterance that included the the word too. Given the previous utterances, this makes sense and it is not present within the alternative results of the non-contextual model. These analysis are, of course, subjective to human perception, but since current automated metrics for

Table 1: OpenSubtitles Dataset Results

	OpenSubtitles 2M				OpenSubtitles 5M			
Model	**Perplexity**	**Average**	**Greedy**	**Extrema**	**Perplexity**	**Average**	**Greedy**	**Extrema**
noCTX	**31.359**	**0.794**	**0.701**	0.887				
CWord	32.926	0.733	0.620	**0.890**	25.855	0.788	0.686	0.886

Table 2: Ubuntu Dialogue Corpus Results

	Ubuntu			
Model	**Perplexity**	**Average**	**Greedy**	**Extrema**
HRED 3-turn	-	0.742	0.524	0.432
VHRED 3-turn	-	**0.777**	0.536	0.448
noCTX	**54.02**	0.658	**0.601**	**0.832**
CWord	54.98	0.632	0.561	0.818

this task have not proven to be reliable enough, human evaluation remains the most solid metric to assess model performance.

7 Conclusion

In this work we presented an alternative way to incorporate contextual information into sequence to sequence for dialog response generation. We did this by adding a small number of extra parameters to the base network architecture .Even though there is no performance improvement in terms of the evaluation metrics considered, there is an important increaes in performance in terms of human evaluation. We also were able to verify that the model actually uses information of the past when in a conversation setting.

We emphasise the need of new automatic ways to evaluate this systems since the actual standard metrics do not correlate properly with human perception.

From the experience working with this models and the research performed the firest thing we notice is that currently there is no a clear way to implement neural conversational models in a complete unsupervised way that can mantain a human-like conversation with a person. Even the top implementations rely in several handcrafted rules and do not achieve convincing results.

Due to data sparsity, and the high number of possible sentences that can be a suitable response to a given input sentence, it would be interesting to try to apply other methods to this problem other than sequence to sequence models, such as adversarial learning for example.

Collaboration Statement

All members contributed equally to the development of the project.

References

Alan Ritter, Colin Cherry, B. D. (2011). Data-driven response generation in social media. Empirical Methods in Natural Language Processing (EMNLP).

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Banchs, R. E. (2012). Movie-DiC: a movie dialogue corpus for research and development. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 203–207. Association for Computational Linguistics.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014a). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*. arXiv: 1406.1078.

Cho, K., van Merrienboer, B., Gulcehre, C., Bah-

danau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*. arXiv: 1406.1078.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual LSTM (CLSTM) models for Large scale NLP tasks. *arXiv:1602.06291 [cs]*. arXiv: 1602.06291.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *arXiv:1506.08909 [cs]*. arXiv: 1506.08909.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nio, L., Sakti, S., Neubig, G., Toda, T., and Nakamura, S. (2014). Conversation dialog corpora from television and movie scripts. In *Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA), 2014 17th Oriental Chapter of the International Committee for the*, pages 1–4. IEEE.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. (2016a). Multiresolution recurrent neural networks: An application to dialogue response generation. *arXiv preprint arXiv:1606.00776*.

Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2015a). A Survey of Available Corpora for Building Data-Driven Dialogue Systems. *arXiv:1512.05742 [cs, stat]*. arXiv: 1512.05742.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2015b). Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *arXiv:1507.04808 [cs]*. arXiv: 1507.04808.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. (2016b). A hierarchical latent variable encoder-decoder model for generating dialogues. *CoRR*, abs/1605.06069.

Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. *CoRR*, abs/1503.02364.

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Simonsen, J. G., and Nie, J.-Y. (2015a). A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion. *arXiv:1507.02221 [cs]*. arXiv: 1507.02221.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015b). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *arXiv:1506.06714 [cs]*. arXiv: 1506.06714.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Vinyals, O. and Le, Q. (2015). A Neural Conversational Model. *arXiv:1506.05869 [cs]*. arXiv: 1506.05869.

Wang, T. and Cho, K. (2015). Larger-Context Language Modelling. *arXiv:1511.03729 [cs]*. arXiv: 1511.03729.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.