# Alexa Socialbot

Carter Brown Abhimanyu Gupta Sumner Hearth Charles Howland
Ishaan Jhaveri George Li Julian Moraes

Cornell University
October 28, 2016

## 1 Overview

The overall design of our conversational module is to break different aspects of the conversational requirements into one of two sections: CC (chit-chat) or DD (data-driven). These reflect our view of the two principal components of a conversation: phrasing responses and incorporating relevant data. The CC module is responsible for creating cohesive sentences in response to the user. The DD section is responsible for finding relevant data and facts as required for formulating a response.

## 2 CC Overview

This module, called "Chit Chat" (CC), generates context-sensitive responses to user queries to engage users and drive continued discourse. The model builds off basic seq2seq models such as [11] to more sophisticated architectures such as the DCGM [10], TACNTN [13], vector embeddings of users' personas and sentiments [6], and experimenting with reinforcement learning in the network training [17, 12, 7]. There is a local DMN [5, 14] to keep track of recent information in the conversation that can be called quickly. Furthermore, this component will provide a vector embedding framework for phrasing responses when we must query the DD module and inject information into the conversation.

## 3 CC Architecture

Text first runs through a classifier to determine whether a proper response to the query requires information "outside" the conversation, i.e. real-world info not recently mentioned in the conversation. This can be accomplished by a mix of classical NLP classifiers coupled with a search through the local user DMN present in the module for a relevant answer (see Figure 1). If more information is required, then it is passed onto the DD module.
In both cases, the flow of information returns to the CC module to be merged together. If classic information retrieval among tuples a la Freebase API is effective, then we will return that answer as has been solved in [2, 15, 16]. On the other hand, accessing a response from our DD memory bank in Figure 2, requires more innovative integration of information.

This is where our choice of vector embedding will affect our design. This module of the program will be one of two possibilities. 1) an in-house embedding utilizing PSMM [8] in conjunction with a specialized LSTM encoder-decoder network ending with a softmax. 2) a form of the TACNTN [13] that is based off a CNN architecture and will provide a different means to capture word order in word sequences.

Additionally, a context-sensitive and user-focused model must take user idiosyncrasies and preferences into account when generating responses. To this end, we propose injecting representative user persona and sentiment vectors into the hidden layer computation of our DCGM.

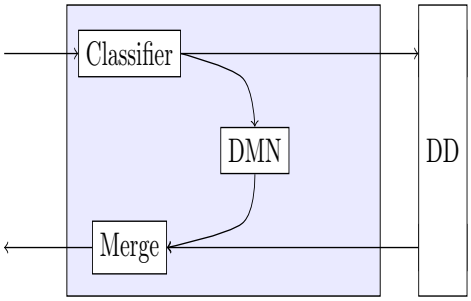


Figure 1: CC architecture

## 4 DD Overview

The Data Driven module (DD) incorporates outside data and information into the conversation. The module assumes that the user's query has been classified in the CC module to require additional information that the standard conversational net cannot provide. DD makes use of recent advances in Neural Network memory systems and topic classification.

A bank of Dynamic Memory Networks (DMNs) [5, 14] stores information on various topics that the user can query. A Topic Classifier (TC) chooses which DMN to use, and will be implemented using SVMs[3, 9], LDA[1, 19], or possibly CNNs[4]. Note that the architecture is "black-boxed" and specific implementation details will be determined through iterative testing (see Methodology).

# 5 DD Architecture

The architecture of the Data Driven (DD) section is given by Figure 2. The topic nodes represent Dynamic Memory Networks (DMNs) [5, 14], and the TC node is a topic classifier to dispatch commands to one of the provided topics.

After the preprocessing of the previous module, English text from the user's query is passed into the TC. The TC determines the topic, i.e. some class, of the user's query. The classes are predefined from training and this section is black-boxed for the sake of the general architecture. However, the topic models proposed in [19] provide an interesting and potentially fruitful framework for classification on short text segments. Furthermore, for each of these classes from training is a DMN+ [14] containing relevant information that we also obtained from training. This training can again be black-boxed for the sake of the design, and different avenues we will investigate include querying Google, Wikipedia, or other databases.

Furthermore, there are DMN+'s on current event topics, such as pop culture, politics, etc. The DMN+'s are continuously updated with information. This captures the idea that important information persists through the DMN+ from day-to-day. Since there has been no research on how large the DMN+'s can effectively be, we will have to determine this once we get to training and implementing. To accommodate the issue, we propose making our topic classifier more fine-grained. If this is not reasonable, then augmenting DMN+'s with more traditional, "non-neural" QA architectures, such as those found in [2, 15, 16], may resolve most issues. However, all of these would require a slight implementation change on a technical scale since these architectures require Freebase API which has a different underlying representation of the data.

Some work (for example [18]) has brought together a knowledge base with a neural net to produce QA responses. However, our approach differs in encapsulating the attention mechanism within the DMN+ bank. Furthermore, while [18] proposes a method to solve the Out of Vocabulary (OOV) problem, it does not appear to accommodate this problem on queries from the user. To remedy this, we propose overlaying the DMN+ with a Pointer Sentinel Mixture Model (PSMM) [8]. Since the PSMM can augment any recurrent architecture with a softmax layer we can augment the answer module described in DMN+ [14] which is in turn fed into a GRU. Since the PSMM has been merged only with LSTM's before, this new architecture feature should make our DMN+ information retrieval bank more robust.

## 5.1 Topic and DMN Distribution

The Topic DMN+'s can be separated into two groups: Global modules and User modules. In Figure 2, the left side and right side are two different levels of distribution–local, and global, respectively.

Global modules are populated with information deemed relevant to all users' interests, such as sports, politics, and culture. These modules can be queried by any DD module for any user and the same responses should be expected (i.e. "Who won the game on Saturday" is not user-dependent). As such these modules can be shared among all instances of the DD module.

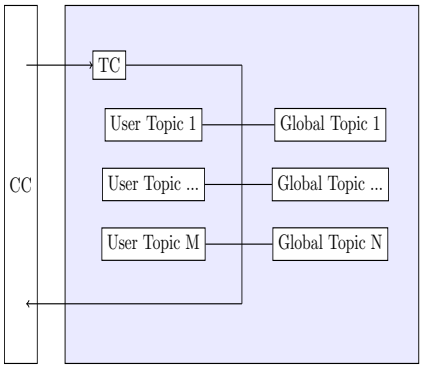


Figure 2: DD architecture

User modules are tailored to a specific user's interests, stored locally for the user. The TC module can double as a simple unsupervised learning system as well as a labeled topic classifier in order to determine what topics a specific user may find interesting. Hence, these classifications evolve with the user as they interact more with Alexa. These topics can then be researched and added to the list of available topic DMNs.

# 6 Information Gathering

Populating the DMNs is a constant process. The general implementation will be to import facts on a topic via predefined news sources (i.e. Google News) or databases (i.e. Wikipedia) and to convert any fact tables into simple sentences. This means that information such as the tuple *(Country: United States, Capital: Washington D.C.)* should be entered into the DMN as *"The capital of the United States is Washington D.C."*. Global topics should be constantly querying for the most relevant information, weighted by importance and age. More important topics should remain in the DMN longer, and more recent topics should be included. Should a query not find results in the topic's DMN, then more classical Information Retrieval algorithms can be employed to compute the result and populate a user DMN for later reference. User DMNs will have a twofold responsibility. They will cover topics not relevant to the general population or topics which are more specific than generally necessary. These user topics can be populated during a conversation, but also when the user is not active. Using the pre-existing TC module we can query incoming news or database entries for information deemed relevant and important, and populate the appropriate DMN for later conversations.

# References


[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, pages 601–608, 2001.

[2] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, Hal Daumé III, Hal Daum, and HD III. A Neural Network for Factoid Question Answering over Paragraphs. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 633–644, 2014.

[3] Thorsten Joachims. Text Categorizaiton with Support Vector Machines: Learning with Many Relevant Features. *Machine Learning*, 1398(LS-8 Report 23):137–142, 1998.

[4] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751, 2014.

[5] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *Nips*, 2015.

[6] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Persona-Based Neural Conversation Model. *Acl*, page 10, 2016a.

[7] Jiwei Li, Will Monroe, Alan Ritter, and Dan Jurafsky. Deep Reinforcement Learning for Dialogue Generation. *arXiv*, (2), 2016b.

[8] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. 2016.

[9] István Pilászy. Text categorization and support vector machines. *The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, 1, 2005.

[10] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and William B. Dolan. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *Naacl-2015*, pages 196–205, 2015.

[11] Orioi Vinyals and Quoc V. Le. A Neural Conversational Model. *arXiv*, 37, 2015.

[12] Jason D. Williams and Geoffrey Zweig. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv*, 2016.

[13] Yu Wu. Response Selection with Topic Clues for Retrieval-based Chatbots.

[14] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic Memory Networks for Visual and Textual Question Answering. *arXiv*, 2016.

[15] Xuchen Yao, Jonathan Berant, and Benjamin Van Durme. Freebase QA: Information Extraction or Semantic Parsing? *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 82–86, 2014a.

[16] Xuchen Yao and Benjamin Van Durme. Information Extraction over Structured Data: Question Answering with Freebase. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 956–966, 2014b.

[17] Wojciech Zaremba and Ilya Sutskever. Reinforcement Learning Neural Turing Machines. *Arxiv*, pages 1–14, 2015.

[18] Yuanzhe Zhang, Kang Liu, Shizhu He, Guoliang Ji, Zhanyi Liu, Hua Wu, and Jun Zhao. Question Answering over Knowledge Base with Neural Attention Combining Global Knowledge Information. *ArXiV*, 2016.

[19] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and Traditional Media using Topic Models. *Proceedings of the 33rd European conference on Advances in information retrieval (ECIR'11)*, pages 338–349, 2011.