

# Chatbot Session

## Hierarchical Attention Networks for Generative Question-Answering Neural Conversational Models

Alexandre Aussem

LIRIS UMR 5205 CNRS  
Data Mining & Machine Learning Group (DM2L)  
University of Lyon 1  
Web: [perso.univ-lyon1.fr/alexandre.aussem](http://perso.univ-lyon1.fr/alexandre.aussem)

January 29, 2019



# Outline

Introduction to neural conversational models

Recurrent neural networks

Learning latent representations

Word embeddings

Single-turn conversation

- The Seq2Seq model

- Attention mechanism

- Evaluation

Multi-turn conversation

- Hierarchical Attention Networks

References

# Introduction to neural conversational models

# Outline of the talk

- ▶ Modern neural-based models for statistical machine translation (SMT) and dialogue systems,
- ▶ Recurrent neural networks (RNN) and **Seq2Seq principles**, a powerful idea in which two RNN work together to transform one sequence to another. An encoder network condenses an input sequence into a vector, and a decoder network unfolds that vector into a new sequence.
- ▶ Elman RNN, GRU and LSTM units and how to implement these models in the Seq2Seq framework,
- ▶ **Word embedding** and Word2Vec principles,
- ▶ The **attention mechanism** which lets the decoder learn to focus over a specific parts of the input sequence or past dialog. The attention mechanism has become one of the hottest topics in deep learning recently.
- ▶ Extension to multi-turn response generation: A hierarchical Seq2Seq with hierarchical attention mechanism for contextual question answering.

# Neural conversational models

- ▶ In recent years, conversational systems have become commonplace. Every day, millions of people use natural-language interfaces such as Siri, Google Now, Cortana, Alexa and others via in-home devices, phones, or messaging channels such as Messenger, WhatsApp, Slack, Skype, among others.
- ▶ Google, IBM, Oracle, Amazon etc. now propose a chatbot builder. Google launched a tool called **Dialogflow** for building chatbots and other conversational applications, with built-in support for speech recognition. They can seamlessly connect to **messaging channels, web environments and social networks** to make scaling easy and to gain users quickly.

## Recent applications and platforms

- ▶ Chatbots are becoming more commonplace on platforms like Facebook messenger, Slack, Whatsapp and even within company websites and apps. Some brands are leveraging chatbots to enhance customer experiences, while others are deploying them in-house.
- ▶ Chatbots helps with instant answers 24/7, increases consultations requests, registrations, online sales, generate more leads for sales team, transforms every facet of every industry and dramatically improve the customer experience
- ▶ Big companies are ready to pay a fee in return for getting 24/7 support, service-level agreement and terms of service that promise data protection, among other things.

# Neural conversational models

- ▶ At the same time, plenty of interesting and impactful contributions that often serve as both a benchmark and an inspiration for new ML methods (NIPS, ICML, ECML, etc.).
- ▶ Challenge tasks are proliferating, including the sixth Dialog Systems Technology Challenge (DSTC6), the Amazon Alexa prize, and the Conversational Intelligence Challenge live competition at NIPS 2017.
- ▶ Artificial datasets are freely available from the bAbI project of Facebook AI Research for testing automatic text understanding and reasoning of end-to-end dialog systems in the restaurant domain, or performing dialog centered around the topic of movies (question answering, recommendation and discussion).

# Neural conversational models

- ▶ From Siri to Google Translate, deep neural networks have enabled breakthroughs in machine understanding of natural language.
- ▶ Most of these models treat language as a **flat sequence of words** or characters, and use a kind of model called a **recurrent neural network** (RNN) to process this sequence (Lowe et al., 2015, Serban et al., 2016, Wen et al., 2015, Li et al., 2016, Sordoni et al., 2015).
- ▶ On a noisy open-domain dataset, these models can perform simple forms of common sense reasoning but still lack of consistency.

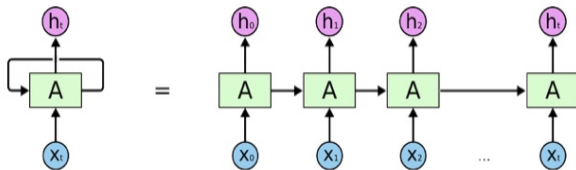


# Neural conversational models

- ▶ A simple approach for this task uses the **sequence to sequence framework** (Seq2Seq) and converses by predicting the next sentence given the previous sentence or sentences in a conversation.
- ▶ It can be trained end-to-end given a large conversational training data set and thus requires **much fewer hand-crafted rules**.
- ▶ Despite optimizing the "wrong" objective function, this straightforward model is able to converse well.
- ▶ On a domain-specific IT helpdesk dataset, it can find a solution to a technical problem via conversations.

# Recurrent neural networks

# Sequence learning with recurrent networks



**An unrolled recurrent neural network.**

Figure from colah.github.io "Understanding LSTM Networks"

- A loop allows information to be passed from one step of the network to the next, allowing information to persist.

# Sequence learning with recurrent networks

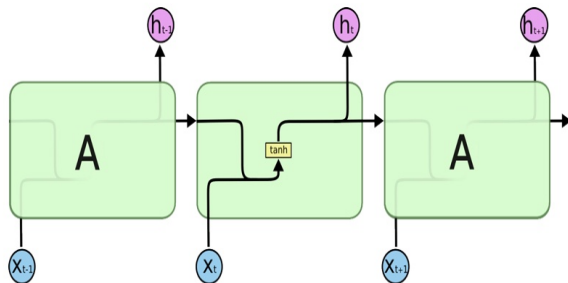


Figure from colah.github.io "Understanding LSTM Networks"

- Such RNN models are not always capable of handling long-term dependencies (i.e. when the time-gap between the relevant information and the moment where it is needed becomes very large).

# Elman RNN

- ▶ Applying a multi-layer Elman RNN with tanh or ReLU non-linearity on an input sequence.
- ▶ For each element in the input sequence, each layer computes the following function:

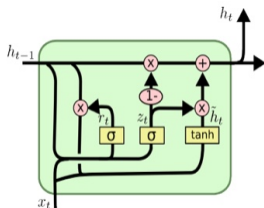
$$h_t = \tanh(w_{ih} * x_t + b_{ih} + w_{hh} * h_{(t-1)} + b_{hh})$$

- ▶  $h_t$  is the hidden state at time  $t$ , where  $h_{t-1}$  is the hidden state at time  $t-1$ , and  $x_t$  is the input of the previous layer at time  $t$  or input  $t$  for the first layer.

# Vanishing gradient

- ▶ The vanishing gradient problem affects many-layered feedforward networks that use backpropagation and also recurrent neural networks.
- ▶ As errors propagate from layer to layer, they shrink exponentially with the number of layers, impeding the tuning of neuron weights that is based on those errors, particularly affecting deep networks.
- ▶ Gated Recurrent Units (GRU) and Long short-term memory (LSTM) networks are RNNs that avoid the vanishing gradient problem. Such networks prevent backpropagated errors from vanishing or exploding.
- ▶ Learning from events that happened far in the past is possible at the expense of more sophisticated neurons. They are augmented by recurrent gates called forget gates that controls the gradient flow.

# Gated Recurrent Unit



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure from colah.github.io "Understanding LSTM Networks"

- ▶ A standard variant of the LSTM is the multi-layer Gated Recurrent Unit (GRU, Cho, et al. 2014). The forget and input gates are merged into a single "update gate." It also merges the cell state and hidden state.
- ▶ The resulting model is simpler than standard LSTM models, and has been increasingly popular.

# LSTM

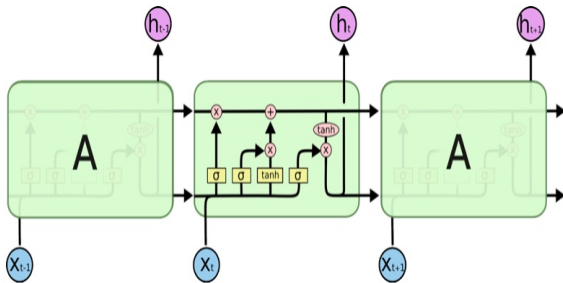


Figure from colah.github.io "Understanding LSTM Networks"

- ▶ Long Short Term Memory networks (LSTM) are a special kind of RNN, capable of learning long-term dependencies (Hochreiter et al. 1997). There are two gates: the forget and input gates that controls the information flow and the memory.



# LSTM

- ▶ LSTM RNN applied to an input sequence. For each element in the input sequence, each layer computes the following function:

$$\begin{aligned}i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}) \\f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}) \\g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg}) \\o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}) \\c_t &= f_t * c_{(t-1)} + i_t * g_t \\h_t &= o_t * \tanh(c_t)\end{aligned}$$

- ▶  $h_t$  is the hidden state,  $c_t$  is the cell state,  $x_t$  is the hidden state of the previous layer at time  $t$ , and  $i_t, f_t, g_t, o_t$  are the input, forget, cell, and out gates, respectively.

# Learning latent representations

# Deep convolutional generative adversarial networks

- ▶ Learning reusable feature representations from large unlabeled datasets has been an area of active research.
- ▶ In computer vision, one can leverage the practically unlimited amount of unlabeled images and videos to learn good intermediate representations, which can then be used on a variety of supervised learning tasks such as image classification.
- ▶ Good image representations can be built by (GANs) (Goodfellow et al., 2014), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks

# Walking in this latent space

- ▶ Walking in this latent space results in semantic changes to the image generations (such as objects being added and removed), the model has learned relevant and interesting representations

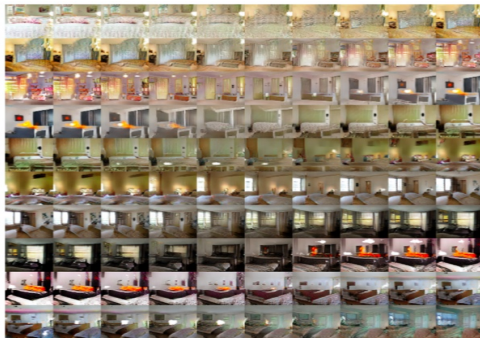


Figure from (Radford, Metz, Chintala, ICLR 2016)

- ▶ Every image in the space plausibly looking like a bedroom (e.g. TV slowly being transformed into a window).

# Vector arithmetic on face samples

A similar structure emerges in the latent representation of the generators.

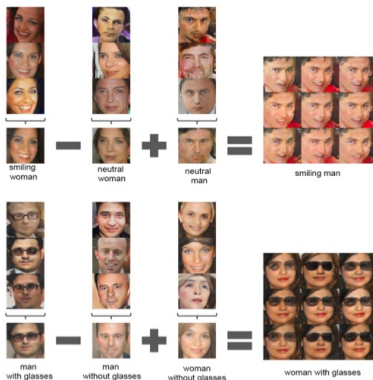


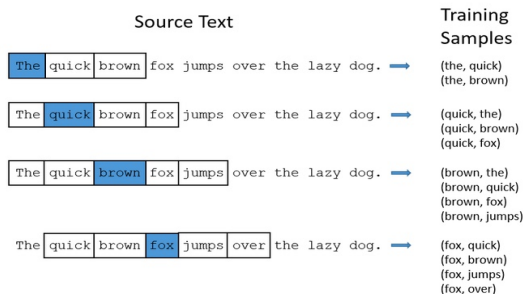
Figure from (Radford, Metz, Chintala, ICLR 2016)

# Word embeddings

# Text preprocessing

- ▶ Tag named entities using a NER tagger (NLTK or Spacy) and replace them with placeholders.
- ▶ Split the corpus into three non-overlapping partitions: train, validation and test.
- ▶ Construct a vocabulary by considering only the words that occur at least 10 times in the corpus. Ignore or replace all the remaining words with an unknown token `< UNK >`.
- ▶ Then learn jointly semantic representations for one-hot encoded words along with the parameters, using the training dataset from scratch, or initialize them with representations learnt off-line.
- ▶ Publicly available Word2vec embeddings (Mikolov et al., 2013) trained on a corpus of 100M words capture rich notions of semantic relatedness useful for language models. Embedding dimension is typically between 100 and 700.

# Skip gram neural network architecture for Word2Vec

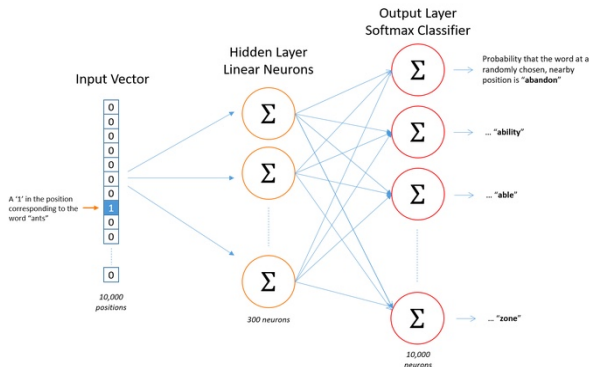


McCormick, C. (2016). Word2Vec Tutorial - The Skip-Gram Model

- ▶ Train the neural network to infer the probability for every word picked at random in our vocabulary of being the “nearby word”, in small window size of 2.
- ▶ A typical window size might be 5, meaning 5 words behind and 5 words ahead (10 in total).



# Skip gram neural network architecture for Word2Vec



McCormick, C. (2016). Word2Vec Tutorial - The Skip-Gram Model

- The output of the (linear) hidden layer is just the “word vector” for the input word. So the end goal is just to learn this hidden layer weight matrix – the output layer is ignored.

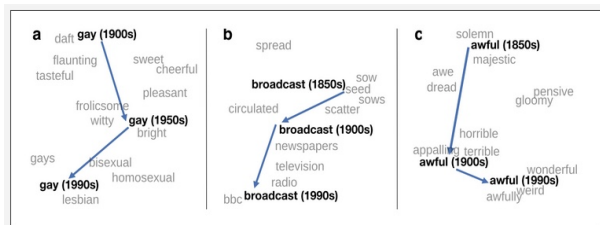
## Word2Vec in practice

- ▶ In the context of evaluating learned representations of words simple arithmetic operations reveal rich linear structure in representation space (Mikolov et al., 2013).
- ▶ We use word similarities obtained from frWaC: a 1.6 billion word corpus constructed from the Web limiting the crawl to the .fr domain and using medium-frequency words from the 'Le Monde Diplomatique' corpus (Dim = 700, Cutoff = 100, trained with CBOW (Fauconnier JP., French Word Embeddings, <http://fauconnier.github.io>, 2015)).

# Vector arithmetic on our samples

- ▶ La femme est à l'homme, ce que la reine est au roi
- ▶ La voiture est à l'homme, ce que la voiturette est à la femme
- ▶ La stupidité est à l'intelligence, ce que la grossièreté est à la beauté
- ▶ La passion est à l'ennui, ce que la mari est à la femme
- ▶ La salarié est à l'employeur, ce que la ouvrier est au travailleur
- ▶ L'innovation est à l'entreprise, ce que le numérique est à l'analogique
- ▶ Le commerce est à l'ingénierie, ce que le économie est à la finance
- ▶ Un homme s'intéresse à la politique
- ▶ Une femme s'intéresse à son homme

# Word Embeddings for Historical Text



William L. Hamilton, et al. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. ACL 2016.

- ▶ Historical word vectors may be used to study the semantic evolution of words across to propose two statistical laws that govern the evolution of word meaning
- ▶ Changes in word meaning are visualized by projecting historical word vectors into a 2-D space. "Gay" shifted in meaning over the last century, from meaning "showy" or "cheerful" to denoting "homosexuality".
- ▶ "Broadcast" used to refer to the act of throwing seeds, but then this motion became associated with the throwing of newspapers, and eventually broadcast developed its current meaning of "disseminating information."
- ▶ "Awful" underwent a process known as pejoration; it used to literally mean "full of awe", but over time it became more negative and now signifies that something is "upsetting."

## Single-turn conversation

Introduction to neural conversational models

Recurrent neural networks

Learning latent representations

Word embeddings

Single-turn conversation

The Seq2Seq model

Attention mechanism

Evaluation

Multi-turn conversation

Hierarchical Attention Networks

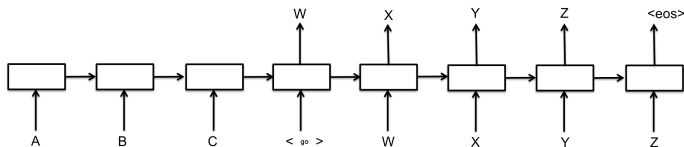
References

# Seq2Seq in Statistical Machine Translation

- ▶ A neural machine translation system is a neural network that directly models the conditional probability  $p(y|x)$  of translating a source sentence,  $x_1, \dots, x_n$ , to a target sentence,  $y_1, \dots, y_m$ .
- ▶ A basic form of NMT consists of two components:
  1. An encoder which computes a representation  $s$  for each source sentence,
  2. A decoder which generates one target word at a time
- ▶ Single-turn generative conversation is similar to SMT. It transforms the source sentence or question into a target answer.

# Standard Seq2Seq

A common practice to build a chatbot is to learn a response generation model within an encoder-decoder framework from large scale message-response pairs (Shang 2015, Vinyals 2015).





# RNN Encoder

- ▶ The RNN Encoder–Decoder framework for SMT (Cho et al. 2014, Sutskever et al. 2014) learns to align and translate simultaneously.
- ▶ In the Encoder–Decoder framework, an encoder reads the input sentence, a sequence of vectors  $\mathbf{x} = (x_1, \dots, x_{T_x})$  into a vector  $c$ .

$$h_t = f_e(h_{t-1}, x_t) \text{ and } c = q(h_1, \dots, h_{T_x})$$

- ▶ where  $h_t$  is the RNN hidden unit,  $c$  a vector generated from the sequence of the hidden states,  $f_e$  and  $q$  are nonlinear functions.
- ▶ Sutskever et al. (2014) used an LSTM as  $f$  and  $q$

$$q(h_1 \dots, h_{T_x}) = h_{T_x}$$

# RNN decoder

- ▶ The probability of decoding each word  $y_j$  is parametrized as:

$$p(y_j|y_{<j}, c) = \text{softmax}(g(s_j))$$

- ▶  $g$  is the transformation function that outputs a vocabulary-sized vector,  $c$  is a vector generated from the sequence of the hidden states.
- ▶  $s_j$  is the decoder hidden unit, abstractly computed as:

$$s_j = f_d(s_{j-1}, c)$$

- ▶  $f_d$  computes the current hidden state given the previous hidden state and can be either a RNN, GRU, or LSTM unit.
- ▶ In earlier works,  $c$  was used once to initialize the decoder hidden state. When  $c$  depends selectively on the encoder hidden states, the approach is referred to as an **attention mechanism** and was proposed in (Bahdanau et al., 2015; Jean et al., 2015).

# RNN decoder

- ▶ The encoder and decoder of the model are jointly trained to maximize the conditional probability of a target sequence given a source sequence.
- ▶ The decoder is often trained to predict the next word  $y_t$  given a iterated context vector  $c_t$  and the previously predicted word.
- ▶ The decoder decomposes the conditional probability as:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^T p(y_t|y_1, \dots, y_{t-1}, \mathbf{x}) = \prod_{t=1}^T p(y_t|y_{t-1}, s_t)$$

where  $s_t$  is an RNN hidden state for time  $t$ , computed by

$$s_t = f_d(s_{t-1}, y_{t-1}, c_t)$$

- ▶ The **context vector**  $c_t$  depends on a sequence of annotations  $(h_1, \dots, h_{T_x})$  to which an encoder maps the input sentence.

Introduction to neural conversational models

Recurrent neural networks

Learning latent representations

Word embeddings

Single-turn conversation

The Seq2Seq model

**Attention mechanism**

Evaluation

Multi-turn conversation

Hierarchical Attention Networks

References

# Attention mechanism

- ▶ A recent trend in Deep Learning are 'attention mechanisms', one of the most exciting advancements according to Ilya Sutskever (research director of OpenAI).
- ▶ With an attention mechanism we **no longer try encode the full source sentence into a fixed-length vector**. Rather, we allow the decoder to “attend” to different parts of the source sentence at each step of the output generation.
- ▶ Importantly, we let the model learn what to **pay more or less attention to the states in the source sentence** while producing each word of the target sentence

# Attention mechanism

- ▶ The context vector  $c_t$  is computed as a weighted sum of these annotations

$$c_t = \sum_{k=1}^T \alpha_{tk} h_k$$

- ▶ The weight  $\alpha_{tk}$  of each annotation  $h_j$  is computed by

$$\alpha_{tk} = \frac{\exp(e_{tk})}{\sum_{p=1}^{T_x} \exp(e_{tp})} \quad \text{where} \quad \exp(e_{tk}) = a(s_{t-1}, h_k)$$

- ▶ The probability  $\alpha_{tk}$ , or its associated energy  $e_{tk}$ , reflects the importance of the annotation  $h_k$  with respect to the previous hidden state  $s_{t-1}$  in deciding the next state  $s_t$  and generating  $y_t$ .

# Attention mechanisms for response generation

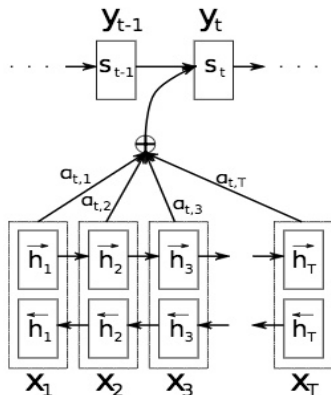


Figure from (Radford, Metz, Chintala, ICLR 2016)

- ▶ **Bidirectional RNN.** Reversing the source sequence (feeding it backwards into the encoder) shortens the path from the decoder to the relevant parts of the encoder.
- ▶ The  $y$ 's are the output words produced by the decoder, and the  $x$ 's are our source sentence words.
- ▶  $y_t$  now depends on a weighted combination of all the input states. The (normalized) weights  $\alpha$  are the contribution of the input states to each output word.

# Attention mechanism

- ▶ The alignment model  $a$  scores how well the inputs around position  $k$  and the output at position  $t$  match. The score is based on the RNN hidden state  $s_{t-1}$  and the  $k$ -th annotation  $h_k$  of the input sentence.
- ▶ The alignment is soft to allow the gradient of the cost function to be backpropagated through.  $a$  is usually modelled as a feedforward neural network which is jointly trained with the encoder/decoder.
- ▶ By letting the decoder have an attention mechanism, it relieves the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector.
- ▶ With this new approach the information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decoder accordingly.



Introduction to neural conversational models

Recurrent neural networks

Learning latent representations

Word embeddings

Single-turn conversation

The Seq2Seq model

Attention mechanism

Evaluation

Multi-turn conversation

Hierarchical Attention Networks

References

# Target loss

- ▶ Open domain dialogue generation aims at generating meaningful and coherent dialogue responses given the dialogue history.
- ▶ Neural conversational systems approximate such a goal by predicting the next dialogue utterance given the dialogue history using the **maximum likelihood estimation** (MLE) objective.
- ▶ Despite its success, this over-simplified training objective leads to problems: responses are dull, generic, repetitive, and short-sighted (Li et al., 2016).
- ▶ Solutions to these problems require answering a few fundamental questions:
  - ▶ What are the crucial aspects that characterize an ideal conversation, how can we quantitatively measure them,
  - ▶ how can we incorporate them into a machine learning system?

# How to evaluate a conversational model?

- ▶ Automatic evaluation of probabilistic language models that generate novel responses remains an open problem of research (Liu et al., 2016)
- ▶ The generated language is sound both syntactically and semantically but its relevance cannot be easily evaluated.
- ▶ Metrics like Perplexity, BLEU, and deltaBLEU, have been adopted from machine translation to understand the performance of such generative models but they are not well suited to dialog systems.
- ▶ We may also consider an answer pool of size  $N$  containing the ground truth and  $N - 1$  other, randomly sampled. The generative model scores all the sentences and rank them according to a scoring function. The  $recall_K$  metric measures the percent of sentences whose ground truth was ranked in the top- $K$  of the answer pool.

# Chatbot of the Master Data Science

Bonjour

→ Bonjour

Je fais un M1 infomatique à Nantes et je suis intéressé par la data science

→ Merci, c'est un bon debut

Quelle est la date de la rentrée ?

→ La rentrée est le 10 septembre à 10h

Quel est le taux de reussite à la sortie de ce M2?

→ Il est de 90%

Quels sont les cours enseignés ?

→ Il faut regarder sur notre site web

Peut-on le faire en alternance ?

→ Non ce M2 n'est pas ouvert en alternance

Quels sont les enseignants du master

→ Des enseignants-chercheurs du Liris

Quelles options choisir en M1 pour se préparer ?

→ Vous devez prendre contact avec le responsable du Master pour en discuter

Merci monsieur

→ de rien

## Multi-turn conversation

# Multi-turn conversation

- ▶ Conversational agents aim to realize natural and human-like conversations with people regarding to a wide range of issues in open domains.
- ▶ A common practice to build a chatbot is to learn a response generation model within an encoder-decoder framework from large scale message-response pairs (Shang 2015, Vinyals 2015).
- ▶ Such models **ignore conversation history** when responding, which is contradictory to the nature of real conversation between humans.
- ▶ To resolve the problem, researchers have taken conversation history into consideration and proposed response generation for multi-turn conversation (Sordoni 2015,2016,Serban 2016).
- ▶ Most tasks in natural language processing, like multi-turn conversation, can be cast into question answering (QA) problems.

# Multi-turn conversation

- ▶ We focus on multi-turn response generation for question answering and task-oriented dialog systems in which we try to learn a relevant response generation model from the responses and **the context**
- ▶ A context refers to a message and several utterances in its previous turns. In practice, when a message comes, the model takes the context as input and generate a response as the next turn.
- ▶ Multi-turn conversation requires a model to generate a response relevant to the whole context. The complexity of the task lies in two aspects:
  1. a conversation context is in a hierarchical structure (words form an utterance, and utterances form the context) and has two levels of sequential relationships among both words and utterances within the structure,
  2. not all parts of the context are equally important to response generation.

# Multi-turn conversation

- ▶ While state-of-the-art methods such as HRED and VHRED (Serban 2016) focus on modeling the hierarchy of the context, the second point is not addressed, although it is often a crucial step for generating a proper response.
- ▶ Without this step, existing models may lose important information in context and generate irrelevant responses.
- ▶ Attention mechanism has raised interest in single-turn response generation (Shang 2015) and multi-turn response generation (Xing 2017, Yang 2016, Sukhbaatar 2015),
- ▶ We discuss a hierarchical recurrent attention network (HRAN) for multi-turn response generation and question answering. The recurrent encoder is similar to HRAN (Yang 2016), with an additional modification to adapt the model to question answering in a task-oriented context.



# BaBI tasks

- ▶ Reading comprehension can be assessed via question answering on the **BaBI goal-oriented and context-dependent restaurant reservation tasks** (Weston 2015, Bordes 2016).
- ▶ The tasks measure understanding in several ways: whether a system is able to answer questions via chaining facts, simple induction, deduction and many more.
- ▶ The tasks are designed to be prerequisites for any system that aims to be capable of conversing with a human.
- ▶ We may check the capacity of generating coherent sentences (language modeling), interpreting a request and asking the right questions to issue an API call as well as the ability to modify an API call.

# BaBI tasks

- ▶ The data is based on an underlying knowledge base, whose facts contain the restaurants that can be booked and their properties.
- ▶ Each restaurant is defined by a type of cuisine (10 choices, e.g., French, Thai), a location (10 choices, e.g., London, Tokyo), a price range (cheap, moderate or expensive) and a rating (from 1 to 8).
- ▶ The KB can be queried using API calls, which return the list of facts related to the corresponding restaurants.
- ▶ Each query must contain four fields: a location, a type of cuisine, a price range and a party size.

## Typical dialog from bAbl task 5

hello  
→ hello what can i help you with today?  
can you make a restaurant reservation in a  
**expensive** price range for **eight**?  
→ i'm on it  
→ any preference on a type of cuisine?  
i love **italian** food  
→ where should it be?  
**in paris**  
→ ok let me look into some options for you  
→ **API\_CALL(italian, paris, eight, expensive)**  
actually i would prefer for **six**  
→ sure is there anything else to update?  
**no**  
→ ok let me look into some options for you  
→ **API\_CALL(italian, paris, six, expensive)**

The task tests: i) the capacity of interpreting a request and asking the right questions to issue an API call, ii) the ability to modify an API call, and iii) the capacity of using outputs from an API call to propose options (sorted by rating) and to provide extra-information (not shown here).

Introduction to neural conversational models

Recurrent neural networks

Learning latent representations

Word embeddings

Single-turn conversation

The Seq2Seq model

Attention mechanism

Evaluation

Multi-turn conversation

Hierarchical Attention Networks

References

# Approach

- ▶ Our model can be cast into the framework of Memory Networks (Weston14), which is by now a classical framework for Question Answering and multi-turn response generation.
- ▶ Standard end-to-end trainable implementations are given in (Sukhbaatar15, Weston16, Bordes16), and several variants were later on developed in the literature, notably
  - ▶ Question Reduction Networks (Seo16),
  - ▶ Dynamic memory Networks DMN and DMN+ of (Kumar16, xiong16),
  - ▶ Hierarchical Recurrent Networks HRED in (Sordoni15, Serban16, Kottur17),
  - ▶ HRAN in (Xing17).
  - ▶ Entity Networks (Henaff16), stores information in a very high-level fashion.

# Approach

- ▶ For multi-turn response generation, models belonging into the class of Memory Networks are typically trained in the following way : Given a user utterance together with the past conversation, the model learns to generate an answer according to a pipeline of four actions :
  1. Storage : The model performs reading and memory storage of the historical part of the conversation.
  2. Input : The user current utterance is acquired, forming a query vector.
  3. Information retrieval : The query vector is used to retrieve relevant information stored in memory, forming a decision vector.
  4. Answer : The decision vector is used to form an answer to the user utterance.

# Approach

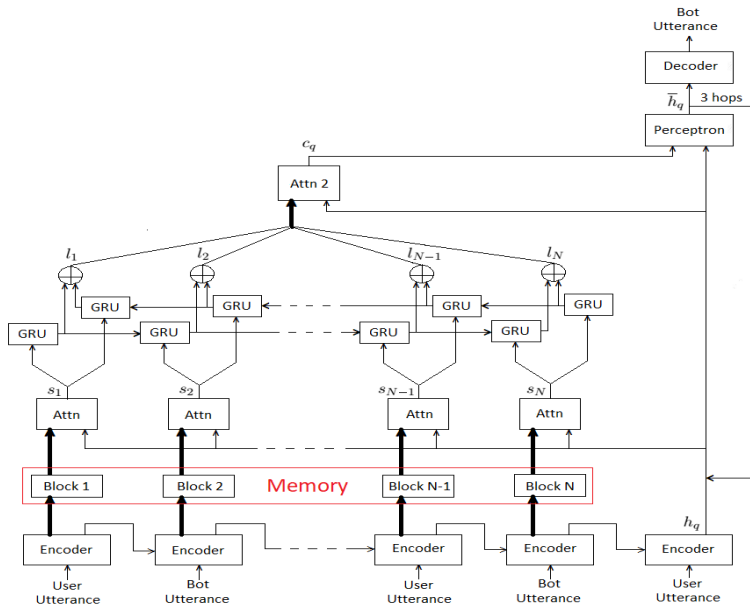
- ▶ The disadvantage of this approach, while it is well suited for the present context, is that the case of large text in the historical part becomes very delicate, as it can be the case for Question-Answering over the content of a book.
- ▶ Problems based on large corpuses implementing a Memory Network can be found in (Chandar16).
- ▶ It is a non-trivial problem to design neural networks that are capable of retrieving information from a large memory while attending only small portions, while being trainable in a simple and efficient way as with Stochastic Gradient Descent.
- ▶ For this type of problems one may consult (Rae2016). An alternative to this approach consists in storing information into a fixed-size memory and repeatedly updating it during conversation with a user, such as in Entity Networks (Henaff16).

# Description of the model

- ▶ Information tracking in memory is done here by two layers of *attention mechanism*. Attention mechanism was first proposed for machine translation (Bahdanau14, Cho14), and is afterwards quickly applied to: single-turn response generation (Vinyals15), language modeling (Yang17), question answering (Bachrach17), multi-turn response generation (Xing17), multi-stream conversational document modeling (Sawada17), multimedia or spoken content comprehension (Tseng16), document classification (Yang16).
- ▶ The information tracker admits a hierarchical structure, where a first treatment is made at the word level and a second at the sentence level. Such approach to deal with text documents and text conversations was widely discussed in literature (Kumar16, Sordoni15, Serban16, Kottur17, Liu17, Zhou17).



# Hierarchical Attention Network



# Word Encoding

- ▶ Given a sentence  $u = w_1, \dots, w_T$ , we first embed the words,  $v_t = W_w \Phi(w_t)$ .
- ▶ We then use a **bidirectional GRU** to get annotations of words by summarizing information from both directions for words.
- ▶ The forward GRU reads the sentence  $v_t$  from  $v_1$  to  $v_T$  and the backward GRU reads from  $v_T$  to  $v_1$ :

$$\begin{aligned}v_t &= W_w \Phi(w_t), \quad t \in [1, T], \\ \vec{h}_t &= \overrightarrow{GRU}(v_t, \vec{h}_{t-1}), \quad t \in [1, T], \\ \overleftarrow{h}_t &= \overleftarrow{GRU}(v_t, \overleftarrow{h}_{t+1}), \quad t \in [T, 1],\end{aligned}$$

- ▶ The **utterance embedding vector**  $h_u$  is set as the concatenation of the last hidden states of the forward and backward GRUs.  $h_u$  is a high level representation of the utterance.

$$h_u = [\vec{h}_T, \overleftarrow{h}_1]$$

# Word Attention

- ▶ Extract from any past utterance the words that are important according to the current user utterance  $q$ , and aggregate the representation of those informative words to form, for each past utterance  $u$ , an **attention vector**  $s_u$ .
- ▶ Given an utterance  $u$  stored in memory as a sequence of annotation vectors  $v_1, \dots, v_T$ , each vector  $v_t$  is given a *score* computed through a simple NN:

$$sc(v_t, h_q) = W^{(2)} \tanh \left( W^{(1)}[v_t, h_q] + b^{(1)} \right)$$

These scores are then used to determine **attention weights**  $\alpha_t \in [0, 1]$  for each vector  $v_t$ , which in turns determine an attention vector  $s_u$  as

$$\alpha_t = \frac{\exp sc(v_t, h_q)}{\sum_{t'} \exp sc(v_{t'}, h_q)} \quad s_u = \sum_{t=1}^T \alpha_t v_t$$

# Sentence Encoding

- ▶ Given the sequence of attention vectors  $s_1, \dots, s_N$  resulting from word attention over the past utterances  $u_1, \dots, u_N$ , we complete each vector with contextual information and form a sequence  $l_1, \dots, l_N$  through another bi-GRU unit:

$$\begin{aligned}\vec{l}_i &= \overrightarrow{GRU'}(s_i, \vec{l}_{i-1}), \quad i \in [1, N], \\ \overleftarrow{l}_i &= \overleftarrow{GRU'}(s_i, \overleftarrow{l}_{i+1}), \quad i \in [N, 1],\end{aligned}$$

where hidden states are initialized at  $\vec{l}_0 = \overleftarrow{l}_{N+1} = 0$ .

- ▶ Again,  $\vec{l}_i$  and  $\overleftarrow{l}_i$  are concatenated to get an annotation of the **attention vector**  $s_i$ , i.e.,  $l_i = [\vec{l}_i, \overleftarrow{l}_i]$ .

# Sentence Attention

- To reward sentences that are clues to correctly answer a question, we again use an attention mechanism to process the sequence  $l_1, \dots, l_N$  and provide a **global context vector**  $c_q$  :  
First a scalar score is computed for each vector  $l_i$  by

$$sc(l_i, h_q) = W^{(4)} \tanh \left( W^{(3)} [l_i, h_q] + b^{(3)} \right)$$

from which attention weights  $\beta_i \in [0, 1]$  are computed, in turns determining the overall context vector  $c_q$  via convex combination as

$$\beta_i = \frac{\exp(sc(l_i, h_q))}{\sum_{i'} \exp(sc(l_{i'}, h_q))} \quad c_q = \sum_{i=1}^N \beta_i l_i$$

The resulting context vector depends selectively on the history of the conversation and summarizes all the relevant information given the current query  $q$ .

# Decision vector

- ▶ The context vector  $c_q$  is then combined with the query vector  $h_q$  through a simple linear operation  $H$

$$h'_q = h_q + Hc_q \quad (1)$$

- ▶ We perform multi-hops in the information tracking process :  
Namely, we again perform word attention and compute a new context vector with  $h'_q$  instead of  $h_q$ , forming a new context vector  $c'_q$  which is then fed jointly with  $h'_q$  in (1), resulting in another vector  $h''_q$ .
- ▶ We perform a third pass as above, finally forming a **decision vector**  $\bar{h}_q$ .

# Answer Decoder

- ▶ The model outputs the conditional probability  $p(u_{N+2}|u_1, \dots, u_{N+1})$  where  $u_{N+1} = q$  (i.e. the current question), given the query vector  $h_q$  and the decision vector  $\bar{h}_q$ . The log likelihood is,

$$\log p(u_{N+2}|u_1, \dots, u_{N+1}) = \sum_{t=1}^T \log (p(w_t|w_{<t}, h_q, \bar{h}_q))$$

- ▶ We fix  $k_0 = \bar{h}_q$  and compute recursively,

$$\begin{aligned} p(w_t|w_{<t}, h_q, \bar{h}_q) &= \text{Softmax}(W_{so}k_t + b_{so}) \\ k_t &= \text{GRU}([w_{t-1}, h_q], k_{t-1}). \end{aligned}$$

- ▶ The GRU updates the current hidden state  $k_t$  and the current word  $w_t$  given  $w_{t-1}, h_q, k_{t-1}$ , starting from the overall context vector, until the EOS token is issued.

# Task-oriented dialog Datasets

- ▶ Synthetic dataset with a list of 20 topics that can be addressed (e.g. courses content, inscription process, deadlines, projects to develop),
- ▶ For each topic a short conversation template is created (from 1 to 5 User-bot exchanges), each sentence being displayed in 2 to 15 different variants and each containing various sub-topics. This allows to generate hundreds of short dialogues using the template of each topic.
- ▶ A list of 30 *conversation templates* is created, each being a short succession of topics, as for instance 1) application details, 2) prerequisites, 3) programming languages, 4) courses content. For each topic in a conversation template a short dialogue is randomly selected,
- ▶ For each dialogue template, 150 dialogues are created for training, 50 for validation and another 50 for testing. This results in 4500 dialogues for training, 1500 for validation and another 1500 for test.



## Task-oriented dialog Datasets

User: Bonjour

Bot : Bonjour, en quoi puis-je vous aider ?

User: Combien y a-t-il de places dans votre master ?

Bot : Quel parcours souhaitez-vous suivre ?

User: Quels sont les parcours proposés ?

Bot : Les parcours informatique et mathématiques

User: Je veux candidater au parcours **informatique**

Bot : Etes-vous inscrit à **Lyon 1** ?

User: **Oui**

Bot : Dans votre cas le nombre de places est  
DB\_Query(**Places, Lyon1, Informatique**)

User: Faut-il avoir un ordinateur personnel ?

Bot : Ce n'est pas nécessaire, mais c'est recommandé

User: Peut-on faire un stage en laboratoire ?

Bot : C'est conseillé si vous souhaitez faire une thèse

User: Et quels sont les **prérequis** pour ce parcours ?

Bot : DB\_Query(**Prerequis, Lyon1, Informatique**)

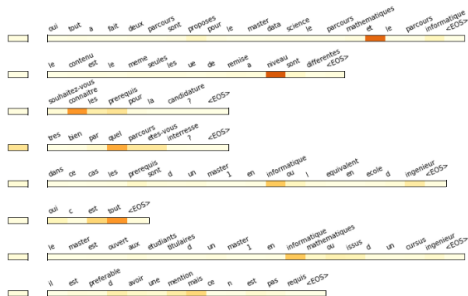
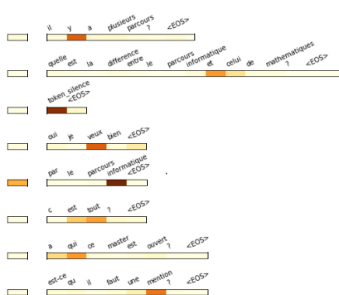
# Task-oriented dialog Datasets

- ▶ There are two decisive features the bot must retrieve when necessary: The country the student is studying and the program he is most interested in (Mathematics or Computer Science).
- ▶ These 2 features impact the inscription process, the contact person, the courses content, the prerequisites . . .
- ▶ A major difficulty for the bot is to retrieve this information when necessary, either from the dialogue history or by directly asking the user.
- ▶ A set of tokens is inserted in the corpus in replacement of certain words that may be subject to future change or for confidentiality reasons.

# Results on the Master's Program dataset

- ▶ With *recurrent* attention modules, the bot is able to perfectly generate 22555 answers out of the 22765 in the corpus, hence having more than 99% rate of success.
- ▶ Among the 210 answers categorized as errors, 163 weren't proper mistakes but actually answers with slight, yet appropriate variations of the ground answer.

# Results on the Master's Program dataset



est-ce qu il y a des options a suivre en master 1 ?

= certaines options du master 1 informatique sont conseillees referez vous a la page siteconseils

< certaines options du master 1 informatique sont conseillees referez vous a la page siteconseils EOS

# Results on the original dialogue bAbI tasks dataset

	Model	
Task 1	99.9	(99.9)
Task 2	99.9	(99.9)
Task 3	<b>98.9</b>	<b>(90)</b>
Task 4	59.3	(3.2)
Task 5	<b>97.8</b>	(66.2)

- ▶ The success on Task 3 shows the model's ability to perform some abstract reasoning, since here the model were supposed to propose restaurant names by decreasing order of their ratings.
- ▶ If the  $i - 1$ -th restaurant suggestion made by the model were discarded by the user, then the  $i$ -th suggestion should be the name of the restaurant whose rating immediately follows.

# References

# References

- ▶ J-B Aujogue and A. Aussem. Hierarchical Recurrent Attention Networks for Context-Aware Education Chatbots. Preprint 2019.
- ▶ Yoram Bachrach, Andrej Zukov Gregoric, Sam Coope, Ed Tovell, Bogdan Maksak, JoseRodriguez, and Conan McMurtie. An attention mechanism for answer selection using a combined global and local view. CoRR, abs/1707.01378, 2017.
- ▶ Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.
- ▶ Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. CoRR,abs/1605.07683, 2016.
- ▶ Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. Hierarchical memory networks. arXiv preprint arXiv:1605.07427, 2016.
- ▶ Kyunghyun Cho, Bart van Merriënboer, Fethi Bougares, HolgerSchwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.
- ▶ Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. arXiv preprint arXiv:1612.03969, 2016.
- ▶ Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. arXiv preprint arXiv:1803.03067, 2018.
- ▶ Satwik Kottur, Xiaoyu Wang, and Vitor R. Carvalho. Exploring personalized neural conversational models. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, pages 3728–3734. AAAI Press, 2017.
- ▶ 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. In International Conference on Machine Learning, pages 1378–1387, 2016.
- ▶ Bing Liu, et al. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. CoRR,abs/1711.10712, 2017.
- ▶ Fei Liu and Julien Perez. Gated end-to-end memory networks. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 1–10, 2017.
- ▶ Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025, 2015.

# References

- ▶ Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. Query-reduction networks for question answering. arXiv preprint arXiv:1606.04582, 2016.
- ▶ Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. Multi resolution recurrent neural networks: An application to dialogue response generation. CoRR, abs/1606.00776, 2016.
- ▶ Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural net-work models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., pages 3776–3784, 2016.
- ▶ Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. CoRR, abs/1605.06069, 2016.
- ▶ Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Si-monsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 553–562. ACM, 2015.
- ▶ Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. Weakly supervised memory networks. CoRR, abs/1503.08895, 2015.
- ▶ Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In Advances in neural information processing systems, pages 2440–2448, 2015.
- ▶ Bo-Hsiang Tseng, Sheng-syun Shen, Hung-yi Lee, and Lin-Shan Lee. Towards machine comprehension of spoken content: Initial TOEFL listening comprehension test by machine. CoRR, abs/1608.06378, 2016.
- ▶ Oriol Vinyals and Quoc V. Le. A neural conversational model. CoRR, abs/1506.05869, 2015.
- ▶ Jason Weston. Dialog based language learning. arXiv preprint arXiv:1604.06045, 2016.
- ▶ Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. CoRR, abs/1502.05698, 2015.
- ▶ Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. 2014.



# References

- ▶ Chen Xing, Wei Wu, Yu Wu, Ming Zhou, Yalou Huang, and Wei-Ying Ma. Hierarchical recurrent attention network for response generation. CoRR, abs/1701.07149, 2017.
- ▶ Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In International conference on machine learning, pages 2397–2406, 2016.
- ▶ Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer, and Alex Smola. Neural machine translation with recurrent attention modeling. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 383–387, 2017.
- ▶ Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016, pages 1480–1489, 2016.
- ▶ Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. End-to-end offline goal-oriented dialog policy learning via policy gradient. CoRR, abs/1712.02838, 2017.
- ▶ Jack Rae, Jonathan J Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Tim Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. In Advances in Neural Information Processing Systems, pages 3621–3629, 2016.
- ▶ Naoki Sawada, Ryo Masumura, and Hiromitsu Nishizaki. Parallel hierarchical attention networks with shared memory reader for multi-stream conversational document classification. In INTERSPEECH, 2017.