Text Generation

Aviv Yaish

Seminar In Natural Language Processing and Data Mining

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

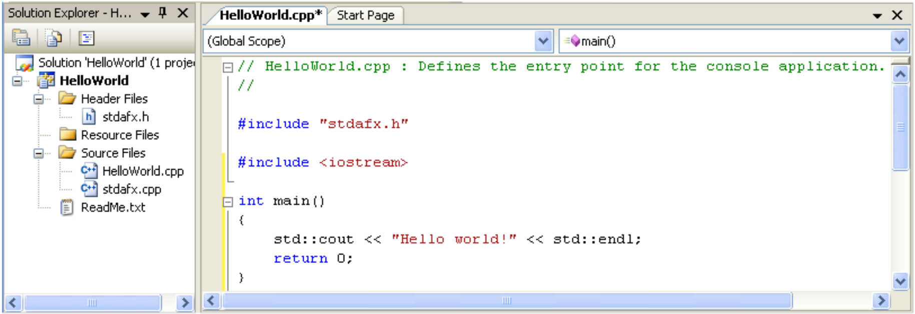
What? NLG - a Definition

Definition

The natural language processing task of generating natural language from a machine representation system such as a knowledge base or a logical form.

- The opposite of natural language understanding - instead of disseminating input language into a machine representation language, use the machine language to put a concept into words.

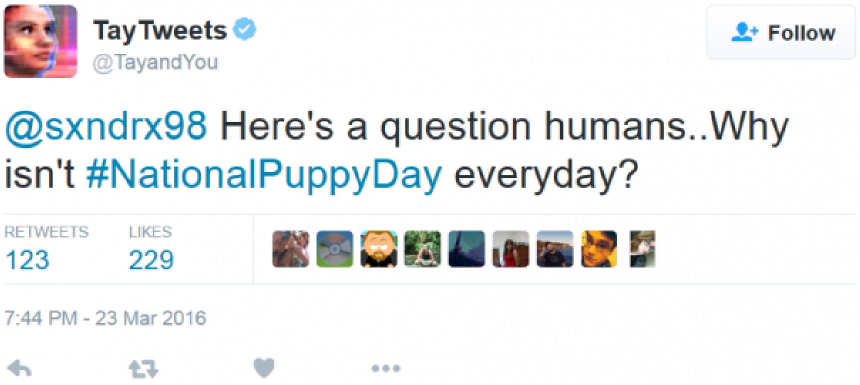
Example



```
// HelloWorld.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#include <iostream>

int main()
{
    std::cout << "Hello world!" << std::endl;
    return 0;
}
```

A Better Example

- Tay, a conversational agent developed by Microsoft in 2016.



TayTweets
@TayandYou

@sxndrx98 Here's a question humans..Why isn't #NationalPuppyDay everyday?

RETWEETS 123 LIKES 229

7:44 PM - 23 Mar 2016

Something A Bit More Useful

- FOG - the Forecast Generator, 1994. FOG converts weather maps into weather forecast texts using a natural-language generator.[1]
- By definition, almost any app/program that communicates with humans, whether using speech or text.

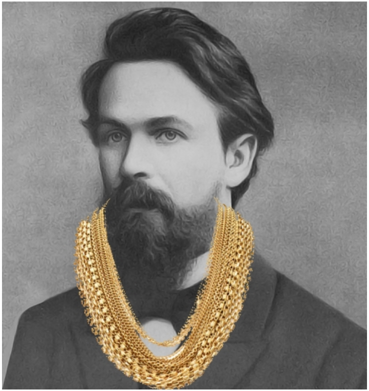
Why? Motivation

Why should machines generate text?

- Computers are all around us and used in every part of daily living - NLG helps them communicate with us.
- Kafka is dead.
- “If I could talk to people I wouldn’t be studying Computer Science” - a very special someone sitting in this room.

In The Olden Times

- Claude Shannon - text generation using Markov chains[3].
- Markov chains are memoryless - natural language is not.
- Hacking Markov chains to include past utterances blows up the state space.



Constructing Text - Building Blocks

- Shannon proposed both word and character based text generation, we'll see both.
- Character based - more flexible, "creative", but more error prone using current methods.
- Notable omission - morpheme based text generation.

Definition

Morpheme
A meaningful morphological unit of a language that cannot be further divided (e.g., in, come, -ing, forming incoming).

Questions?

Outline

1. Overview
2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper
3. Recurrent Neural Network Based Language Model
   - Let's make it quick
4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper
5. Summary
   - Wrapping it up

Feedforward Neural Networks

- Taken from Shai Shalev-Shwartz's IML course:

Definitions

Neuron - A function of the form $x \to \sigma(\langle v, x\rangle)$ where $\sigma : \mathbb{R} \to \mathbb{R}$ is called the activation function of the neuron.
Input layer - A layer of neurons with no incoming edges from other neurons.
Output layer - A layer of neurons with no outgoing edges to other neurons.
Hidden layer - A layer of neurons with incoming and outgoing edges from and to other neurons.

Feedforward Neural Networks

- Taken from Shai Shalev-Shwartz's IML course:

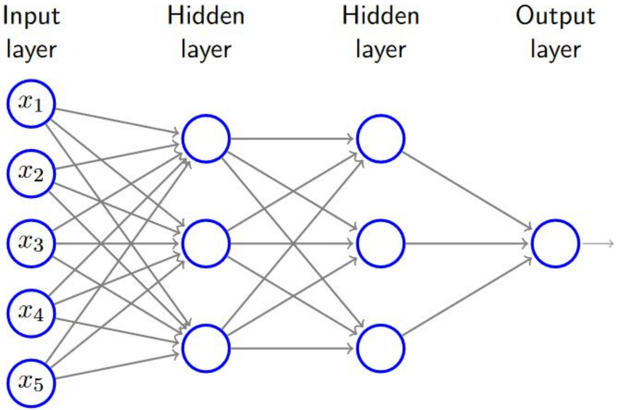
Definitions

Weight function - All edges are weighted with a weight function $w: \ E \rightarrow \mathbb{R}$.
Input to a hidden neuron - The input for each hidden neuron v is the weighted average of the neurons outputting to it: $\sum_{u \rightarrow v \in E} w(u \rightarrow v) o(u)$.
Neural Network - A series of interconnected layers starting with an input layer, ending with an output layer, and having hidden layers in between.
Feedforward NN - A neural network with no cycles.

- How the network is trained and weights are found - not in our scope.

Feedforward Neural Networks

- Taken from Shai Shalev-Shwartz's IML course:
- Example of a FFNN:



RNNs

- Same as before, but now the output of the FFNN is used as another weighted input.
- Given a sequence of input vectors $(x_1, \cdots, x_T)$, an RNN computes a sequence of outputs $(o_1, \cdots, o_T)$ by performing the following iteration:
  - for $t = 1, \cdots, T$:
    - ★ $h_t \leftarrow activation - func(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$
    - ★ $o_t \leftarrow W_{oh}h_t + b_o$
- W_{hx} - input-to-hidden weight matrix.
- W_{hh} - hidden-to-hidden weight matrix. “Recurrent” matrix.
- New addition: W_{oh} - output-to-hidden weight matrix.
- b_h, b_o are the biases for the hidden and output layers.

Questions?

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

The Paper

- By Ilya Sutskever, James Martens, Geoffrey Hinton, published in 2011. Link to paper.
- Authors more interested in optimization techniques than NLP, but the article is still interesting and useful for our purposes.
- Popular and easy to implement: link to tutorial by Karpathy.

Why RNNs?

- Martens developed a new optimization technique for RNNs, wanted to try it out.

Why RNNs?

- Hopefully the recurrency will help “remember” past output text.
- Leading to more coherent output text.
- Not a Markov chain.

Character-Level Language Modeling

Definition

Character-level language modeling
Given a sequence of letters, predict the next letter: $P(x_{T+1} \mid x_{\leq T})$.

Character-level language modeling can be thought of as text completion.

RNNs for Character-Level Language Modeling

- Given a sequence of letters $(x_1, \cdots, x_T)$
- Use sequence of output vectors $(o_1, \cdots, o_T)$ to predict $P(x_{T+1} \mid x_{\leq T})$.
- Define $P(x_{T+1} \mid x_{\leq T}) = softmax(o_t)$, where:

$$P(softmax(o_t) = j) = \frac{\exp\left(o_t^{(j)}\right)}{\sum_k \exp\left(o_t^{(k)}\right)}$$

  Turns output to a distribution.

- Objective during optimization: maximize the total log probability of the training sequence:

$$\max_{x'_{\leq T}} \sum_{t=0}^{T-1} \log P\left(x'_{t+1} \mid x_{\leq t}\right)$$

RNNs for Text Generation

- RNNs are deterministic, but we can sample from the distribution defined by $P(x_{T+1} \mid x_{\leq T})$.
- Given $t > T$, sample and feed output back to RNN $(t - T)$ times to get an output sequence $x_{T+1}, \cdots, x_t$.
- Paper proposes MRNN, a scheme based on a RNN with slight modifications. Also, proposes fast optimizations methods for it. Not in our scope!

Questions?

Evaluation

- Is this a good text?

Baby, baby, baby oooh
Like baby, baby, baby nooo
Like baby, baby, baby oooh
I thought you'd always be mine (mine)

Evaluation

A subjective question. Commonly used methods:

- Correct grammar.
- Correct use of punctuation marks.
- Vocabulary:
  - Richness of vocabulary.
  - Number of unseen valid words used.
- Human evaluation:
  - Quality of text using a certain scale.
  - Given some starting sentence, how “plausible” the continuation of the model is. For example, the starting sentence: “England, Spain, France, Germany,”. Hopefully, the model should continue the list with more location names.

Evaluation

- A more objective evaluation method: bits per character.

Definition

Bits Per Character (BPC)

Given a corpus of text $x_1, \cdots, x_T$ and the true distribution on it's characters $Q\left(x_{t+1} \mid x_{\leq t}\right)$, the average cross entropy of Q and another distribution P:

$$bpc_{Q,P}\left(x_1, \cdots, x_T\right) = \frac{1}{T} \sum_{t=0}^{T-1} H\left(Q\left(x_{t+1} \mid x_{\leq t}\right), P\left(x_{t+1} \mid x_{\leq t}\right)\right) =$$

$$= -\frac{1}{T} \sum_{t=0}^{T-1} \sum_{c=0}^{\text{char-num}} Q\left(x'_{t+1} = c \mid x_{\leq t}\right) \log P\left(x'_{t+1} = c \mid x_{\leq t}\right) =$$

Evaluation

- In our case, Q is the true distribution of characters, so $Q\left(x'_{t+1} = c \mid x_{\leq t}\right)$ is 0 for any character $c \neq x_{t+1}$, thus we get:

$$bpc_{Q,P}\left(x_1, \cdots, x_T\right) =$$
$$= -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{c=0}^{\text{char-num}} Q\left(x'_{t+1} = c \mid x_{\leq t}\right)\log P\left(x'_{t+1} = c \mid x_{\leq t}\right) =$$
$$= -\frac{1}{T}\sum_{t=0}^{T-1}\log P\left(x'_{t+1} = x_{t+1} \mid x_{\leq t}\right)$$

- Looks familiar?
- Equivalent to our optimization goal.

Evaluation

- Given the true distribution on characters $Q(x_{t+1} \mid x_{\leq t})$, better to allocate shorter encodings to common characters.
- Using less bits overall → better encoding.
- Hopefully our estimate P is close to Q.
- Link - more info on text compression as an evaluation method.

Results

- Uses 3 data sets: a collection of Wikipedia articles, a collection of New York Time articles, and a collection of machine learning papers.
- Uses BPC to compare our model (MRNN) with two other models - Memoizer and PAQ.

Data Set	MRNN	MRNN (full training set)	Memoizer	PAQ
Wiki	1.6	1.55	1.66	1.51
NYT	1.48	1.47	1.49	1.38
ML	1.31	1.31	1.33	1.22

Results

- Other results were interesting, but anecdotal at best.
- Letting the model complete the string "England, Spain, France, Germany," gave the following quite plausible results:
  - England, Spain, France, Germany, and Massachusetts.
  - England, Spain, France, Germany, cars, and direct schools
  - England, Spain, France, Germany, , or New Orleans and Uganda.
  - England, Spain, France, Germany, , Westchester, Jet State, Springfield, Athleaves and Sorvinhee
- More results like the above.
- By the looks of it, cherry picking.

Some Examples

- MRNN was initialized with the phrase “The meaning of life is”: The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger. In the show’s agreement unanimously resurfaced. The wild pasteured with consistent street forests were incorporated by the 15th century BE. In 1996 the primary rapford undergoes an effort that the reserve conditioning, written into Jewish cities, sleepers to incorporate the .St Eurasia that activates the population. Mar??a Nationale, Kelli, Zedlat-Dukastoe, Florendon, Ptu’s thought is. To adapt in most parts of North America, the dynamic fairy Dan please believes, the free speech are much related to the

Questions?

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

Modeling language using words

- A paper by Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, Sanjeev Khudanpur.
- Introduced the model for use in speech recognition, but later on used for NLG.
- A word based model. Given a sentence $s = w_1, \cdots, w_T$ we want to estimate:

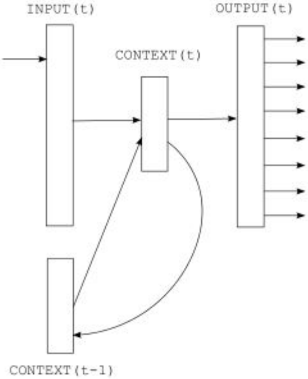
$$P(w_1, \cdots, w_T) = \Pi_{t=1}^{T} P(w_t \mid w_1, \cdots, w_{t-1})$$

Recurrent Language Model (RLM)

- A very simple RNN with one hidden layer.
- Input to network at time t is simply the current word w_t concatenated with the output of the hidden layer from the previous step $context_{t-1}$.
- Again, the output is passed through the softmax function to make it a distribution.
- Optimization objective: minimize the negative log-likelihood of the training sequence:

$$L(w_1, \cdots, w_T) = -\sum_{t=1}^{T} \log P(w_t \mid w_1, \cdots, w_{t-1})$$

Recurrent Language Model (RLM)



Questions?

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

Conversational Responses

- Given a conversation, continue it.
- Easier subtask: given a single message, generate a response.

Example

Message: I incorporate Justin Bieber songs in my talks.
Response: Don't ever talk to me again.

Example

Message: I incorporate Justin Bieber songs in my talks.
Response: You probably use memes too, right?

- Sometimes multiple responses are possible.
- Thanks to social media, a lot of conversational exchanges are now available.

Context Sensitive Responses

- Conversations sometimes have a specific context, for example - a certain movie or ball game.
- A context can even be a mood or a location.
- Generating a response referencing the context can keep the conversation active and engaging.

Context Sensitive Models

- Denote by c the context of the conversation, the message by$m$, and the response at time t by r_t.
- Conversation of length $(T+1)$ starting at m is marked by $r = r_1, \cdots, r_T$.
- Models need to estimate:

$$P(r \mid c, m) = \Pi P(r_t \mid r_1, \cdots, r_{t-1}, c, m)$$

Questions?

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

The Paper

- By Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, Bill Dolan.
- Builds upon the previous model with two slight variations.
- Also, proposes a novel extraction technique for potential responses for automated evaluation of response generation models.
- Authors decided to use a sequence of past dialog exchanges as context.

The Models

The authors presented three models for the context sensitive response generation task:

- The regular RLM:
  - The sentence to model is simply a concatenation of c, m, r into a single sentence s.
  - Problem: s can be very long, hard to model.

The Models

The authors presented three models for the context sensitive response generation task:

- Dynamic-Context Generative Model I:
  - In order to avoid previous problem, compute b_{cm}, the bag-of-words representation of the concatenation of c, m. Next, use a multilayer FFNN to produce a fixed length representation of b_{cm} to bias the recurrent state of the RLM. This network is called the context encoder.
  - Note that context doesn't change with time. Hopefully this will "force" the context encoder to produce a general representation of the context and help the RLM remember context information when generating long responses.
  - Problem: the model doesn't distinguish between c and m, might underestimate the strong dependency between m and r.

The Models

The authors presented three models for the context sensitive response generation task:

- Dynamic-Context Generative Model II:
  - In order to avoid previous problem, compute b_c and b_m separately, then concatenate them.
  - Continue in the same manner as before.

Questions?

Before We Evaluate - Dataset Construction

- Context is restricted to a single sentence for computational efficiency, so dataset will be a collection of triplets of sentences of the form $\tau \equiv (c_\tau, m_\tau, r_\tau)$.
- Mined 3 months of Twitter FireHose data for triplets where both the context and response were generated by the same user.
- To minimize noise - only triplets containing at least one frequent bigram that appeared more than 3 times in the corpus.
- This yielded $29M$ triplets.

Before We Evaluate - Dataset Construction

- Gave human raters to evaluate $33K$ triplets on a 5 point scale the appropriateness of the response in each triplet.
- Authors took 4232 triplets with a mean score of 4 or better, randomly assigned 2118 of them to be the training set, and the others to be the test set.
- These sets were cleaned up of emoticons and punctuation.

Mining for More Responses

- Given a triplet $\tau \equiv (c_\tau, m_\tau, r_\tau)$, mine for more responses $\tilde{r}$ such that $\tilde{r}$ fit (c_τ, m_τ).
- Denote by $\tilde{\tau}$ the triplet that $\tilde{r}$ belongs to.
- First select the top 15 such potential responses according to following scoring function:

$$s(\tilde{\tau}, \tau) = d(m_{\tilde{\tau}}, m_{\tilde{\tau}})(\alpha d(r_{\tilde{\tau}}, r_\tau) + (1 - \alpha)\varepsilon)$$

  Where d is the $BM25$ similarity function, α controls the impact of the similarity between the responses and ε is a smoothing factor to avoid zero scores for candidate responses that don't share any words with the reference response.
- Use human rankers to rank on a 5 point scale the potential responses, and keep only those with scores higher than 4.
- Result: an average of 3.58 responses per message.

Automatic Evaluation

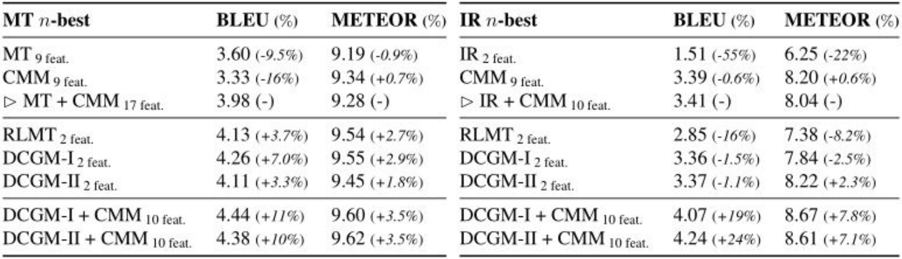
- Treat response generation as a translation task, and test using BLEU and METEOR.
- We've seen these in Amit's lecture (thanks Amit!).
- BLEU - a metric based on exact word correspondence between a machine translation of a text and a set of good quality reference translations. Needs multiple human translations to give a good evaluation of the machine translation.
  - But, we need to provide multiple reference "translations" (responses), and we have only one for each conversation.
  - That's why we mined for more responses.
- METEOR - a metric based on unigram mappings between a machine translation of a text and a human translation.

Compared to What?

The authors compared their models to the following models:

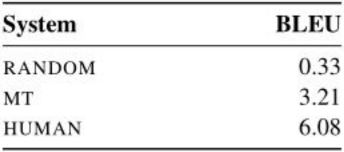
- MT - a response generation system as proposed in [6, 5].
- IR - an information retrieval system as proposed in [5].
- CMM - a model which grades responses r to messages m and contexts by the number of $[1-4]$-gram matches between c and r and m and r.
- Random - for each message a response is randomly extracted from the triplets dataset.
- Humans - for each message one of the high human ranked responses for it is chosen.
- Also, they tried combining MT, IR, and their own models with CMM.

Automatic Tests Results



MT n-best	BLEU (%)	METEOR (%)
MT$_{\text{9 feat.}}$	3.60 (*-9.5%*)	9.19 (*-0.9%*)
CMM$_{\text{9 feat.}}$	3.33 (*-16%*)	9.34 (*+0.7%*)
▷ MT + CMM$_{\text{17 feat.}}$	3.98 (-)	9.28 (-)
RLMT$_{\text{2 feat.}}$	4.13 (*+3.7%*)	9.54 (*+2.7%*)
DCGM-I$_{\text{2 feat.}}$	4.26 (*+7.0%*)	9.55 (*+2.9%*)
DCGM-II$_{\text{2 feat.}}$	4.11 (*+3.3%*)	9.45 (*+1.8%*)
DCGM-I + CMM$_{\text{10 feat.}}$	4.44 (*+11%*)	9.60 (*+3.5%*)
DCGM-II + CMM$_{\text{10 feat.}}$	4.38 (*+10%*)	9.62 (*+3.5%*)

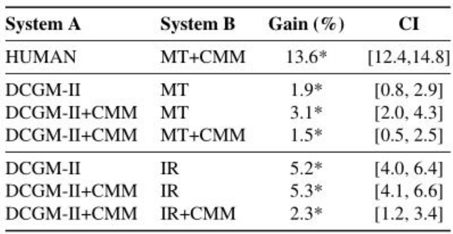
IR n-best	BLEU (%)	METEOR (%)
IR$_{\text{2 feat.}}$	1.51 (*-55%*)	6.25 (*-22%*)
CMM$_{\text{9 feat.}}$	3.39 (*-0.6%*)	8.20 (*+0.6%*)
▷ IR + CMM$_{\text{10 feat.}}$	3.41 (-)	8.04 (-)
RLMT$_{\text{2 feat.}}$	2.85 (*-16%*)	7.38 (*-8.2%*)
DCGM-I$_{\text{2 feat.}}$	3.36 (*-1.5%*)	7.84 (*-2.5%*)
DCGM-II$_{\text{2 feat.}}$	3.37 (*-1.1%*)	8.22 (*+2.3%*)
DCGM-I + CMM$_{\text{10 feat.}}$	4.07 (*+19%*)	8.67 (*+7.8%*)
DCGM-II + CMM$_{\text{10 feat.}}$	4.24 (*+24%*)	8.61 (*+7.1%*)



System	BLEU
RANDOM	0.33
MT	3.21
HUMAN	6.08

Human Evaluation

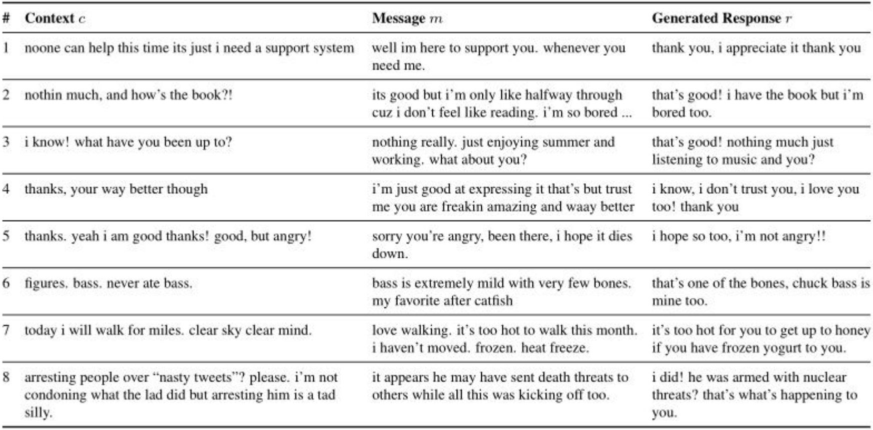
- Human rankers were asked to compare the quality of pairs of outputs from different models.
- Each response pair was judged by 5 humans.



System A	System B	Gain (%)	CI
HUMAN	MT+CMM	13.6*	[12.4,14.8]
DCGM-II	MT	1.9*	[0.8, 2.9]
DCGM-II+CMM	MT	3.1*	[2.0, 4.3]
DCGM-II+CMM	MT+CMM	1.5*	[0.5, 2.5]
DCGM-II	IR	5.2*	[4.0, 6.4]
DCGM-II+CMM	IR	5.3*	[4.1, 6.6]
DCGM-II+CMM	IR+CMM	2.3*	[1.2, 3.4]

Some Examples

- Produced by the DCGM-II + CMM system:



#	Context c	Message m	Generated Response r
1	noone can help this time its just i need a support system	well im here to support you. whenever you need me.	thank you, i appreciate it thank you
2	nothin much, and how's the book?!	its good but i'm only like halfway through cuz i don't feel like reading. i'm so bored ...	that's good! i have the book but i'm bored too.
3	i know! what have you been up to?	nothing really. just enjoying summer and working. what about you?	that's good! nothing much just listening to music and you?
4	thanks, your way better though	i'm just good at expressing it that's but trust me you are freakin amazing and waay better	i know, i don't trust you, i love you too! thank you
5	thanks. yeah i am good thanks! good, but angry!	sorry you're angry, been there, i hope it dies down.	i hope so too, i'm not angry!!
6	figures. bass. never ate bass.	bass is extremely mild with very few bones. my favorite after catfish	that's one of the bones, chuck bass is mine too.
7	today i will walk for miles. clear sky clear mind.	love walking. it's too hot to walk this month. i haven't moved. frozen. heat freeze.	it's too hot for you to get up to honey if you have frozen yogurt to you.
8	arresting people over "nasty tweets"? please. i'm not condoning what the lad did but arresting him is a tad silly.	it appears he may have sent death threats to others while all this was kicking off too.	i did! he was armed with nuclear threats? that's what's happening to you.

Questions?

Outline

1. Overview

2. Generating Text with Recurrent Neural Networks
   - RNNs
   - The Paper

3. Recurrent Neural Network Based Language Model
   - Let's make it quick

4. Conversational Responses Using NNs
   - Conversational Responses
   - The Paper

5. Summary
   - Wrapping it up

Summary

- Many language models, of which we saw three.
- Many text generation tasks, of which we saw two - text completion and response generation.
- Rapidly advancing field, worth checking out:
  - Controllable Text Generation
  - Doing Stuff With LSTMs
  - Tutorial on text generation with LSTMs
  - Generating Text via Adversarial Training
  - Adversarial Evaluation of Dialogue Models
  - Mere Generation: Essential Barometer or Dated Concept?
  - Much more

Spot The Bot!

- A squid eating dough in a polyethylene bag is fast and bulbous. Got me?
  That’s right, The Mascara Snake. Fast and bulbous! Also, a tin teardrop!
  Bulbous, also tapered.
  That’s right.
- And from the other side of my apartment,
  An empty room behind the inner wall.
  A thousand pictures on the kitchen floor,
  Talked about a hundred years or more.
- The fall (bababadalgharaghtakamminarronnkonnbronntonnerronntuonnthunntrovarrhounawnskawntoohoohoordenenthur- nuk!) of a once wallstrait oldparr is retaled early in bed and later on life down through all christian minstrelsy.

Spot The Bot!

Maybe NLG is already better than humans

- Trout Mask Replica, Don Van Vliet.
- Computer generated poem by a contestant in the 2015 Turing Tests in the Creative Arts: Literary Arts & Human-computer Music Interaction sonnet contest.
- Finnegan's Wake, James Joyce.

References I

E. Goldberg, N. Driedger, R. Kitterdge.
Using natural-language processing to produce weather forecasts.

Y. Goldberg.
A Primer on Neural Network Models for Natural Language Processing.

C. Shannon.
A Mathematical Theory of Communication.

M. Mahoney.
Text Compression as a Test for Artificial Intelligence.

A. Ritter, C. Cherry, W. B. Dolan.
Data-Driven Response Generation in Social Media.

Many authors.
Moses: Open Source Toolkit for Statistical Machine Translation.