

# Natural Language Generation

## Natural Language Processing

(based on revision of Abigail See and Antoine Bosselut Lectures)



# Announcement

- TA announcements (if any)...



# Suggested Readings

1. [The Curious Case of Neural Text Degeneration](#)
2. [Get To The Point: Summarization with Pointer-Generator Networks](#)
3. [Hierarchical Neural Story Generation](#)
4. [How NOT To Evaluate Your Dialogue System](#)



# Intro



# Natural Language Generation

**Natural Language Generation (NLG)** refers to any setting in which we generate (i.e., write) new text.

NLG is a subcomponent of

- Neural Machine Translation (NMT)
- (Abstractive) Summarization
- Dialogue (chit-chat and task-based)
- Creative writing: storytelling, poetry-generation
- Free Question Answering (i.e., answer is generated, not extracted from text or knowledge base)
- Image captioning
- ...



# Language Modeling

- **Language Modeling (LM):** the task of **predicting the next word**, given the words so far

$$P(\mathbf{y}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})$$

- A system that produces this probability distribution is called a LM
- If that system is an RNN, it's called an **RNN-LM**



# Conditional Language Modeling

- **Conditional Language Modeling (LM):** the task of **predicting the next word**, given the words so far as well as some input  $\mathbf{X}$

$$P(\mathbf{y}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}, \mathbf{x})$$

- Examples of conditional language modeling tasks:
  - Machine Translation (x = source sentence, y = target sentence)
  - Summarization (x = input text, y = summarized text)
  - Dialogue (x = dialogue history, y = next utterance)
  - ...

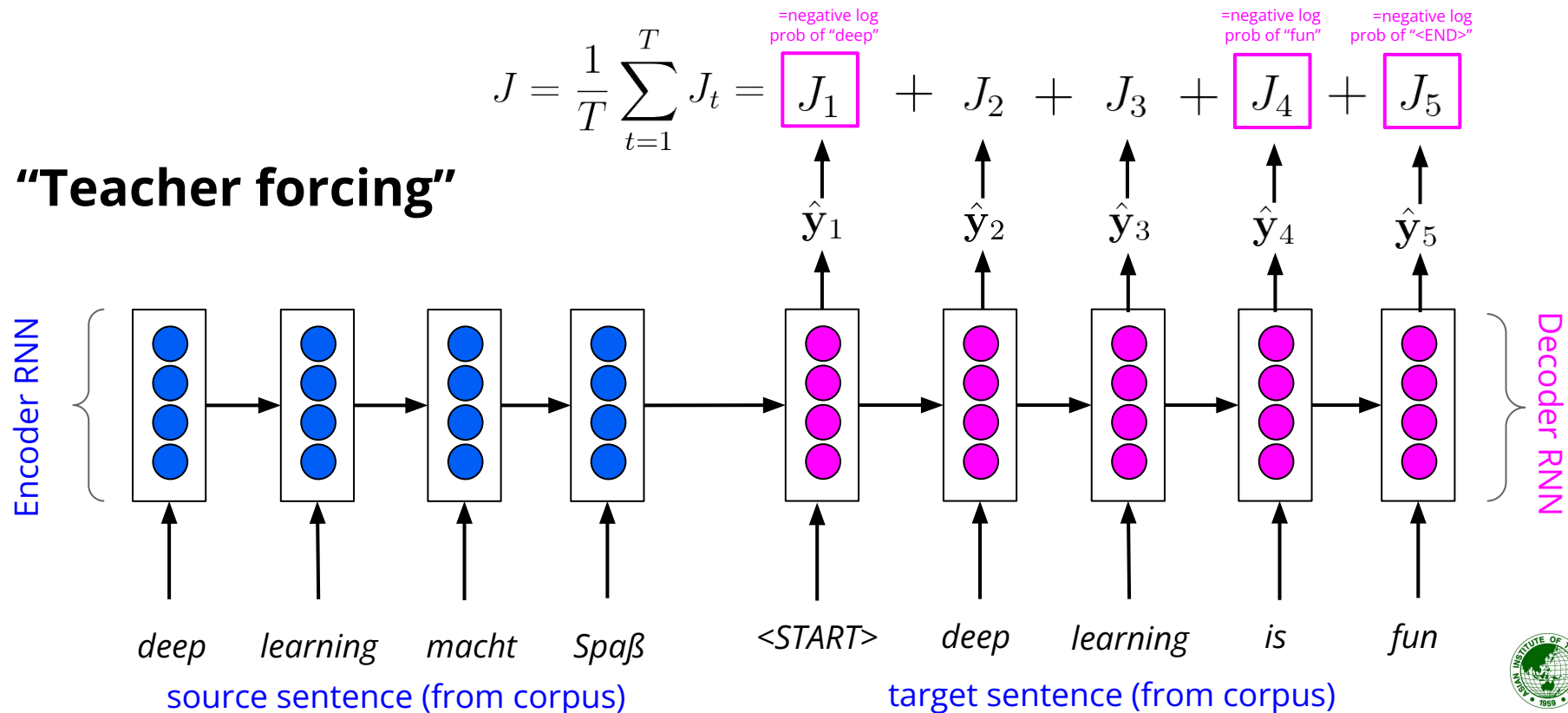


# Training a (conditional) RNN-LM

$$J = \frac{1}{T} \sum_{t=1}^T J_t = \boxed{J_1} + J_2 + J_3 + \boxed{J_4} + \boxed{J_5}$$

=negative log prob of "deep"      =negative log prob of "fun"      =negative log prob of "<END>"

**"Teacher forcing"**





# Decoding algorithms

- How do we use the RNN-LM to generate text?
  - A **decoding algorithm** is an algorithm we use to generate text from our LM
  - We've learned about **two** decoding algorithms:
  - **Greedy decoding**
    - Take argmax (i.e., simply take the word with highest probability)
    - Then feed that as the input on the next step
    - Keep going until you produce <END> or reach some length
    - Usually bad because of lack of backtracking
  - **Beam search**
    - Keep track of the k most probable partial sequences
    - Stop when specified number of hypotheses is reached or when some length is reached
    - Choose the hypothesis with the highest probability (factoring in some adjustment for length)



# What's the effect of beam size (k) ?

- **Small k** has similar problems to greedy decoding (k=1)
  - Ungrammatical, unnatural, nonsensical, incorrect
- **Larger k** means you consider more hypothesis
  - Increasing k is more computationally expensive
  - Increasing k **introduce other problems:**
    - For NMT, **increasing k too much decreases BLEU score (Tu et al., Koehn et al.)**
      - Primarily because it produces too-short translations (even with score normalization!)
        - It can even produce empty translations (Stahlberg & Byrne 2019)
    - In open-ended tasks like chit-chat dialogue, large k can make output more generic (next slide)

Neural Machine Translation with Reconstruction, Tu et al, 2017 <https://arxiv.org/pdf/1611.01874.pdf>  
Six Challenges for Neural Machine Translation, Koehn et al, 2017 <https://arxiv.org/pdf/1706.03872.pdf>



# Effect of beam size in chit-chat dialogue

*I mostly eat a fresh and raw diet, so I save on groceries*



Human  
chit-chat  
partner

| Beam size | Model response                               |
|-----------|----------------------------------------------|
| 1         | <i>I love to eat healthy and eat healthy</i> |
| 2         | <i>That is a good thing to have</i>          |
| 3         | <i>I am a nurse so I do not eat raw food</i> |
| 4         | <i>I am a nurse so I am a nurse</i>          |
| 5         | <i>Do you have any hobbies?</i>              |
| 6         | <i>What do you do for a living?</i>          |
| 7         | <i>What do you do for a living?</i>          |
| 8         | <i>What do you do for a living?</i>          |

## Low beam size:

More on-topic but nonsensical; bad English

## High beam size:

Converges to safe "correct" response, but it's generic and less relevant



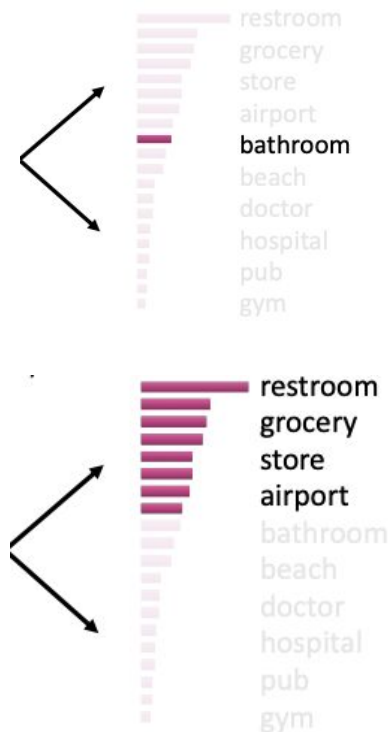
# Sampling-based decoding - another method

- **Pure sampling**

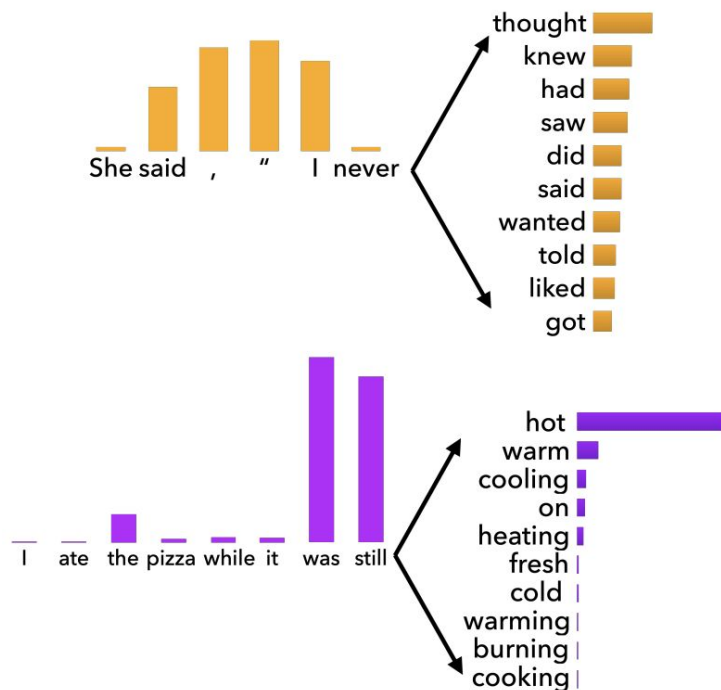
- On each step  $t$ , randomly sample from the probability distribution to obtain your next word
- Like greedy decoding, but sample instead of argmax

- **Top-k sampling**

- (this  $k$  is **NOT** related to beam search)
- On each step  $t$ , randomly sample from the probability distribution, but **restricted to just the top-k most probable words**
- Like pure sampling, but truncate the probability distribution
- $k = 1$  is greedy search,  $k = V$  is pure sampling
- **Increase  $k$**  to get more **diverse/risky** output
- **Decrease  $k$**  to get more **generic/safe** output



# Issues with Top-k sampling



Top-*k* sampling can cut off too ***quickly!***

Top-*k* sampling can also cut off too ***slowly!***

# Sampling-based decoding - another method

- **Top-p sampling**

- On each step  $t$ , randomly sample from probability distribution, but **restricted to just the top-p cumulative distribution mass (i.e., where mass is concentrated)**
- Varies  $k$  depending on the uniformity of the distribution



# Softmax temperature - scaling randomness

- **Recall:** On timestep  $t$ , the LM computes a prob dist  $P_t$  by applying the softmax function to a vector of scores  $\mathbf{s} \in \mathbb{R}^{|V|}$  (this score is generated by our neurons).

$$P_t(w) = \frac{\exp(\mathbf{s}_w)}{\sum_{w' \in V} \exp(\mathbf{s}_{w'})}$$

- We can apply a temperature hyperparameter tau [any number] to the softmax

$$P_t(w) = \frac{\exp(\mathbf{s}_w / \tau)}{\sum_{w' \in V} \exp(\mathbf{s}_{w'} / \tau)}$$

- **Raise the temperature** tau to make the prob dist **more uniform**
  - Thus more diverse output (prob is spread around more vocabs)
- **Lower the temperature** to make prob dist **more spiky**
  - Thus less diverse output (probability is concentrated on top words)
- Note that softmax temperature is **NOT a decoding** algorithm
  - It's a technique you apply at test time, in conjunction with a decoding algorithm (such as beam search or sampling)



# Summary

- **Greedy decoding** is a simple method; gives low quality output
- **Beam search** (especially with high beam size) searches for high probability output
  - Delivers better quality than greedy, but if beam size is too high, can return high probability but unsuitable output (e.g., too generic)
- **Sampling methods** are a way to get more diversity and randomness
  - Good for open-ended / creative generation (poetry, stories)
  - Top-k / Top-p sampling allows you to control diversity
- **Softmax temperature** is another way to control diversity
  - Basically a parameter to scale the confidence level of your model
  - It's not a decoding algorithm! It's a technique that can be applied alongside any decoding algorithm
- Also other methods, for example, **recalibrating** the distributions using existing statistics of n-grams (Khandelwal et al. ICLR 2020), or **re-ranking** the distribution using some outside metric like perplexity or style or anything (Holtzman et al., 2018)
- **Teacher forcing** remains the key but there is a lot of work trying to add diversity (e.g., scheduled sampling, exposure bias, or training with reinforcement learning)
- **A lot more work to be done! Very fun area to explore!**





# Generation Tasks and Approaches - Text Summarization



# Summarization: task definition

**Task:** given input text  $\mathbf{X}$ , write a summary  $\mathbf{Y}$  which is shorter and contains the main information of  $\mathbf{X}$

Summarization can be **single-document** or **multi-document**

- **Single-document** means we write a summary  $\mathbf{Y}$  of a single document  $\mathbf{X}$
- **Multi-document** means we write a summary  $\mathbf{Y}$  of multiple documents  $\mathbf{X}_1, \dots, \mathbf{X}_n$



# Summarization: task definition

Within single-document summarization, there are **datasets** with source documents of **different lengths and styles**:

- **Gigaword**: first one or two sentences of a news article
- **LCSTS**: Chinese microblogging
- **NYT, CNN/DailyMail**: news article
- **Wikihow**: full how-to article
- XSum (Narayan et al., 2018), Newsroom (Grusky et al., 2018), etc.
- ...

**Sentence simplification** is a different but related task: Rewrite the source text in a simpler (sometimes shorter) way

- Simple Wikipedia: standard Wikipedia sentence
- Newsela: news article

List of summarization datasets, papers, and codebases: <https://github.com/mathsyouth/awesome-text-summarization>



# Summarization: task definition

Two main strategies

- **Extractive** summarization: select parts (typically sentences) of the original text to form a summary
  - Easier but restrictive (no paraphrasing)
- **Abstractive** summarization: generate new text using natural language generation techniques
  - More difficult but more flexible



# Pre-neural summarization

- Pre-neural summarization systems were mostly **extractive**. Pipeline:
  - **Content selection**: choose some sentences to include. To select, there are many ways:
    - **Sentence scoring functions** - presence of topic words (computed from tf-idf or the like) or based on features such as position of the sentence
    - **Graph based algorithms**: document as nodes, and edges as connection between each sentence pair; edge weight is proportional to sentence similarity; then use graph algorithms to identify sentences that are **central** to the graph
  - **Information ordering**: choose an ordering of those sentences
  - **Sentence realization**: edit the sequence of sentences (e.g., simplify, remove parts, fix continuity issues)



# Summarization evaluation: Recall BLEU [Papineni et al., ACL 2002]

## BLEU (Bilingual Evaluation Study)

- BLEU **compares** the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - a. **n-gram precision** (usually for 1, 2, 3 and 4-grams)
  - b. Plus a **penalty** for too-short system translations
- BLEU is useful but **imperfect**
  - a. There are **MANY** valid ways to translate a sentence
  - b. So a **good** translation can get a **poor** BLEU score because it has low n-gram overlap with the human translation

BLEU: a Method for Automatic Evaluation of Machine Translation, Papineni et al, 2002. <http://aclweb.org/anthology/P02-1040>



# Summarization evaluation: ROUGE [Lin, ACL 2004]

**ROUGE** (Recall Oriented Understudy for Gisting Evaluation)

- ROUGE not ROGUE (from star wars!)

Like BLEU, it's based on n-gram overlap.

Differences:

- ROUGE has no brevity (too-short) penalty
- ROUGE is based on **recall**, while BLEU is based on **precision**
  - Arguably, **precision** is more important for MT (then add brevity to fix under-translation) and **recall** is more important for summarization (assuming you have a max length constraint)
  - However, often a F1 (combination of precision and recall) version of ROUGE is reported anyway
- Anyway, they are used together to evaluate text summarization...

ROUGE: A Package for Automatic Evaluation of Summaries, Lin, 2004 <http://www.aclweb.org/anthology/W04-1013>



# Summarization evaluation: ROUGE

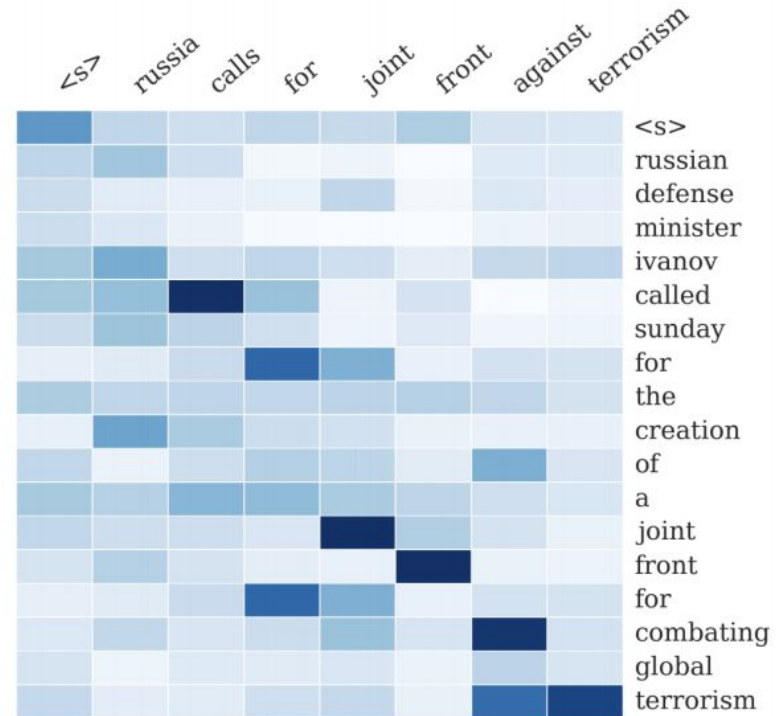
- **BLEU** is reported as a **single number**, which is combination of the precision for  $n=1, 2, 3, 4$  n-grams
- **ROUGE** scores are reported **separately for each n-gram**
- The most commonly-reported ROUGE scores are:
  - ROUGE-1: **unigram** overlap
  - ROUGE-2: **bigram** overlap
  - ROUGE-L: **longest common subsequence** overlap
- A python implementation of ROUGE
  - <https://github.com/pltrdy/rouge>





# Neural summarization - seq2seq attention

- **2015:** Rush et al. publish the first seq2seq summarization paper
- Works on single-document **abstractive** summarization and treat it as a translation task!
- Thus we can apply standard seq2seq + attention NMT methods



A Neural Attention Model for Abstractive Sentence Summarization, Rush et al, 2015 <https://arxiv.org/pdf/1509.00685.pdf>



# Neural summarization: abstractive + extractive [See et al., ACL 2017]

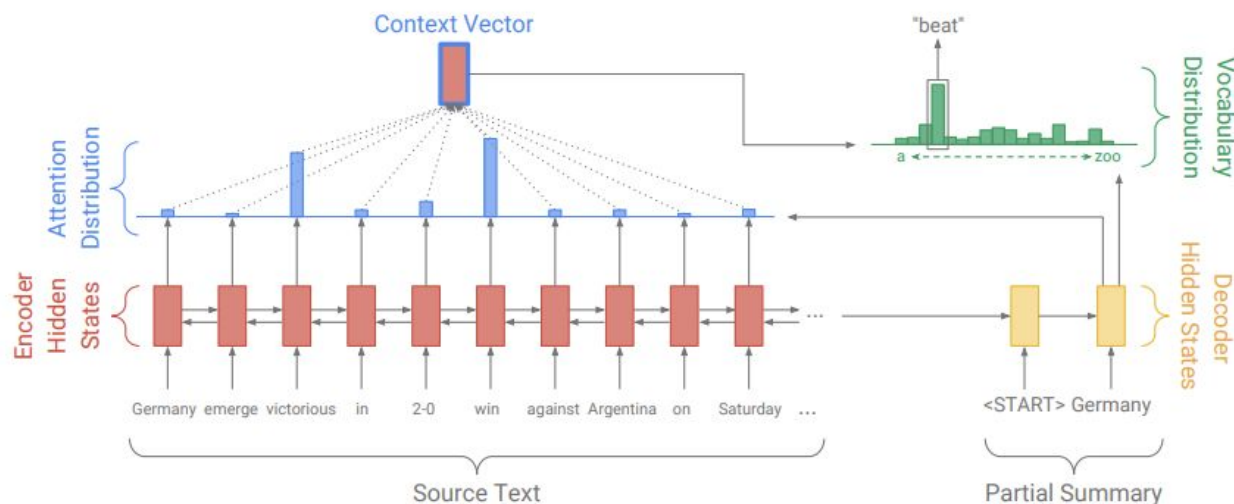
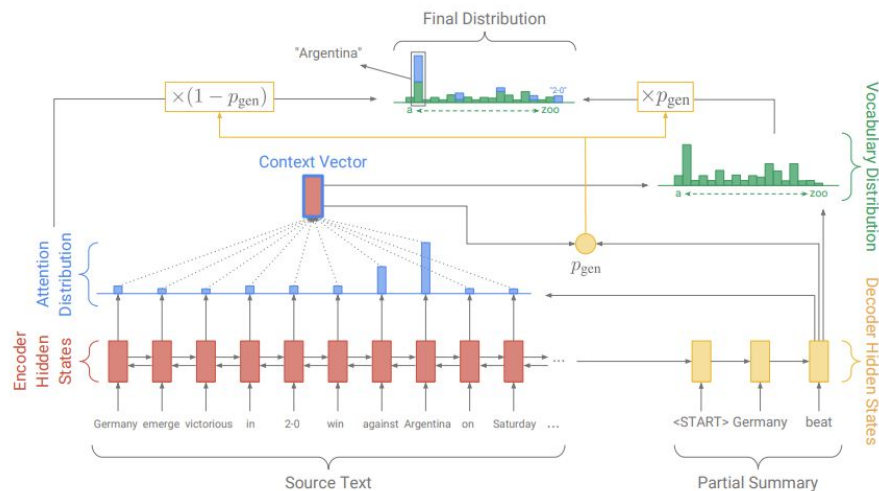


Figure 2: Baseline sequence-to-sequence model with attention. The model may attend to relevant words in the source text to generate novel words, e.g., to produce the novel word *beat* in the abstractive summary *Germany beat Argentina 2-0* the model may attend to the words *victorious* and *win* in the source text.

Get To The Point: Summarization with Pointer-Generator Networks, See et al, 2017, <https://arxiv.org/pdf/1704.04368.pdf>



# Neural summarization: abstractive + extractive



- Combining “copy” and “generation”
- The weight on whether to copy from source or to generate is controlled by  $p_{gen}$
- Here, the equation of the left:  $\mathbf{c}_t$  is the context vector,  $\mathbf{s}_t$  is the decoder state,  $\mathbf{x}_t$  is the decoder input, and the  $\mathbf{a}_t$  is the attention distribution.  $p_{gen}$  is in the range of  $[0, 1]$  because of the sigmoid.  $P_{vocab}$  is the usual softmax function.

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \alpha_i^t$$

$$p_{gen} = \sigma(\mathbf{w}_c^\top \mathbf{c}_t + \mathbf{w}_s^\top \mathbf{s}_t + \mathbf{w}_x^\top \mathbf{x}_t + b)$$

# Neural summarization: copy mechanisms

- Seq2seq+attention systems are good at **generating** fluent output, but **bad at copying over the source**
- **Copy mechanisms** use attention to enable a seq2seq system to easily copy words and phrases from the input to the output
  - Clearly this is very useful for summarization
  - Allowing both copying and generating gives us a hybrid **extractive/abstractive approach**



# Neural summarization: copy mechanisms

- **Many copy mechanisms** are proposed:
  - Language as a Latent Variable: Discrete Generative Models for Sentence Compression, Miao et al, 2016 <https://arxiv.org/pdf/1609.07317.pdf>
  - Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond, Nallapati et al, 2016 <https://arxiv.org/pdf/1602.06023.pdf>
  - Incorporating Copying Mechanism in Sequence-to-Sequence Learning, Gu et al, 2016 <https://arxiv.org/pdf/1603.06393.pdf>



# Neural summarization: copy mechanisms

- **Big problem** with copy mechanisms
  - They copy too much!
    - Mostly long phrases, sometimes even whole sentences
  - What should be an **abstractive** system collapses to a **mostly extractive** system
- **Another problem:**
  - They are bad at overall content selection, especially if the input document is long
  - No overall strategy for selecting content



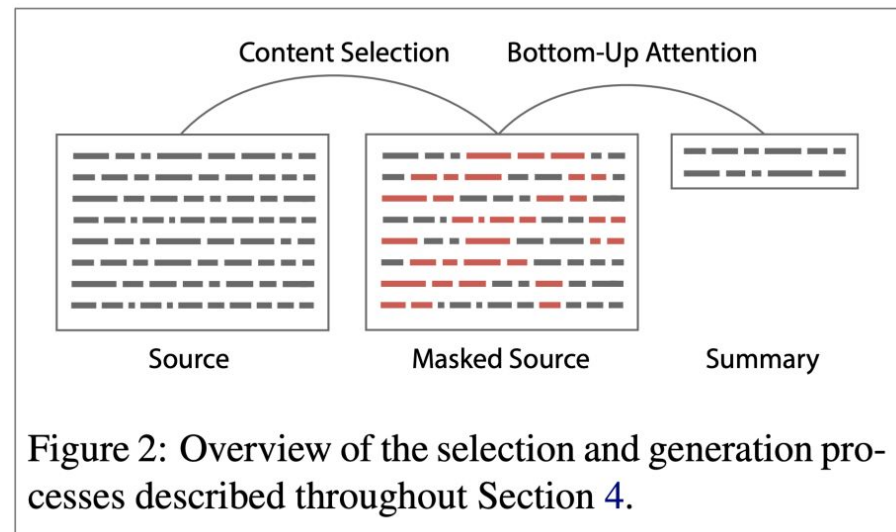
# Neural summarization: better content selection

- **Recall:** pre-neural summarization had **separate** stages for **content selection** and **sentence realization** (i.e., text generation)
- In a standard seq2seq+attention summarization system, these two stages are **mixed** in together
  - On each step of the decoder (i.e., sentence realization), we do word-level content selection (attention)
  - This is bad!: no global content selection strategy
- One solution: bottom-up summarization



## Neural summarization: bottom-up summarization [Gehrmann et al., ACL 2018]

- **Content selection stage:** Use a neural sequence-tagging model to tag words as 'include' or 'don't include' (like POS tagging!)
- **Bottom-up attention stage:** The seq2seq+attention system cannot attend to words tagged as 'don't include' (apply a mask)
  - Very effective! Better global content selection. Less copying of very long sequences





# Neural summarization via Reinforcement Learning [Paulus et al., ICLR 2017]

- In **2017** Paulus et al. published a “deep reinforced” summarization model
- Main idea: Use **Reinforcement Learning (RL)** to directly optimize **ROUGE-L**
  - By contrast, standard maximum likelihood (ML) training cannot directly optimize **ROUGE-L** because it's a **non-differentiable function!**
- Interesting finding:
  - Using RL instead of ML achieved higher ROUGE scores, but lower human judgment scores....LoL
    - But hybrid approach does best!!

| Model                     | ROUGE-1      | ROUGE-2      | ROUGE-L      |
|---------------------------|--------------|--------------|--------------|
| ML, no intra-attention    | 44.26        | 27.43        | 40.41        |
| ML, with intra-attention  | 43.86        | 27.10        | 40.11        |
| RL, no intra-attention    | <b>47.22</b> | 30.51        | <b>43.27</b> |
| ML+RL, no intra-attention | 47.03        | <b>30.72</b> | 43.10        |

**ROUGE scores**

| Model | Readability | Relevance   |
|-------|-------------|-------------|
| ML    | 6.76        | 7.14        |
| RL    | 4.18        | 6.32        |
| ML+RL | <b>7.04</b> | <b>7.45</b> |

**Human judgement**

A Deep Reinforced Model for Abstractive Summarization, Paulus et al, 2017 <https://arxiv.org/pdf/1705.04304.pdf>



# Generation Tasks and Approaches - Dialogue

(Read this on your own time)



# Dialogue

“**Dialogue**” encompasses a large variety of settings:

## Task-oriented dialogue

- **Assistive** (e.g., customer service, giving recommendations, question answering, help user accomplish a task like buying or booking)
- **Co-operative** (two agents solve a task together through dialogue)
- **Adversarial** (two agents compete in a task through dialogue)

## Social dialogue

- **Chit-chat** (for fun or company)
- **Therapy**



# Pre- and post-neural dialogue

- Due to the difficulty of open-ended freeform NLG, pre-neural dialogue systems often used predefined **templates**, or **retrieve** an appropriate response from a **corpus of responses**
- As in summarization research, sine 2015, there have been many papers applying seq2seq methods to dialogue - thus leading to a renewed interest in **open-ended freeform dialogue** systems
- Some early seq2seq dialogue papers include:
  - A Neural Conversational Model, Vinyals et al, 2015  
<https://arxiv.org/pdf/1506.05869.pdf>
  - Neural Responding Machine for Short-Text Conversation, Shang et al, 2015  
<https://www.aclweb.org/anthology/P15-1152>

This is a nice overview of recent (mostly neural) conversational AI work:

<https://medium.com/gobeyond-ai/a-reading-list-and-mini-survey-of-conversational-ai-32fcee97180>



# Seq2seq-based dialogue

- However, it quickly became apparent that a naive application of standard seq2seq+attention methods has serious pervasive deficiencies for (chitchat) dialogue:
  - **Genericness** / boring responses
  - **Irrelevant responses** (not sufficiently related to context)
  - **Repetition**
  - **Lack of context** (not remembering conversation history)
  - **Lack of consistent persona**



# Seq2seq-based dialogue - irrelevant response problem

- **Problem:** seq2seq often generates response that's unrelated to user's utterance
  - Either because it's generic (e.g., "I don't know")
  - Or changing the subject to something unrelated
- **Solution:** optimize for Maximum Mutual Information (MMI) between input  $S$  and response  $T$ :

$$\log \frac{p(S, T)}{p(S)p(T)} = \log \frac{p(T|S)}{p(T)} = \log p(T|S) - \log p(T)$$

$$\hat{T} = \operatorname{argmax}_T \{ \underbrace{\log p(T|S)}_{\text{usual}} - \underbrace{\log p(T)}_{\text{added}} \}$$

This  $p(T)$  is saying  
don't output too  
"generic" stuffs

# Seq2seq-based dialogue - boring response problem

- **Easy test-time fixes**

- Directly upweight rare words during beam search
- Use a sampling decoding algorithm rather than beam search

- **Conditioning fixes**

- Condition the decoder on some additional content (e.g., sample some content words and attend to them)
- Train a retrieve-and-refine model rather than a generate-from-scratch model
  - i.e., sample an utterance from your corpus of human-written utterances, and edit it to fit the current scenario

Why are Sequence-to-Sequence Models So Dull?, Jiang et al, 2018 <https://staff.fnwi.uva.nl/m.derijke/wp-content/papercite-data/pdf/jiang-why-2018.pdf>



# Seq2seq-based dialogue - repetition problem

- **Simple solution:**

- Directly block repeating n-grams during beam search
  - Works well!

- **More complex solutions:**

- Train a coverage mechanism - in seq2seq, this is an objective that prevents the attention mechanism from attending to the same words multiple times
- Define a training objective to discourage repetition
  - If this is a non-differentiable function of the generated output, then will need some technique like e.g., RL to train





# Seq2seq-based dialogue - lack of consistent persona problem

- In 2016, Li et al proposed a seq2seq dialogue model that learns to encode both conversation partners' **personas as embeddings**
  - The generated utterances are conditioned on the embeddings
- There is now a chit-chat dataset called **PersonaChat**, which includes personas (collections of 5 sentences describing personal traits) for every conversation
  - Allow researchers to build persona-conditional dialogue agents

A Persona-Based Neural Conversation Model, Li et al 2016, <https://arxiv.org/pdf/1603.06155.pdf>

Personalizing Dialogue Agents: I have a dog, do you have pets too?, Zhang et al, 2018 <https://arxiv.org/pdf/1801.07243.pdf>




# Negotiation dialogue

In 2017, Lewis et al. collected a **negotiation dialogue dataset**

- Two agents negotiate (via natural language) how to divide a set of items
- The agents have different value functions for the items
- The agents talk until they reach an agreement

Divide these objects between you and another Turker. Try hard to get as many points as you can!  
Send a message now, or enter the agreed deal!

| Items                                                                             | Value | Number You Get                 |
|-----------------------------------------------------------------------------------|-------|--------------------------------|
|  | 8     | <input type="text" value="1"/> |
|  | 1     | <input type="text" value="1"/> |
|  | 0     | <input type="text" value="0"/> |

Fellow Turker: I'd like all the balls

You: Ok, if I get everything else

Fellow Turker: If I get the book then you have a deal

You: No way - you can have one hat and all the balls

Fellow Turker: Ok deal

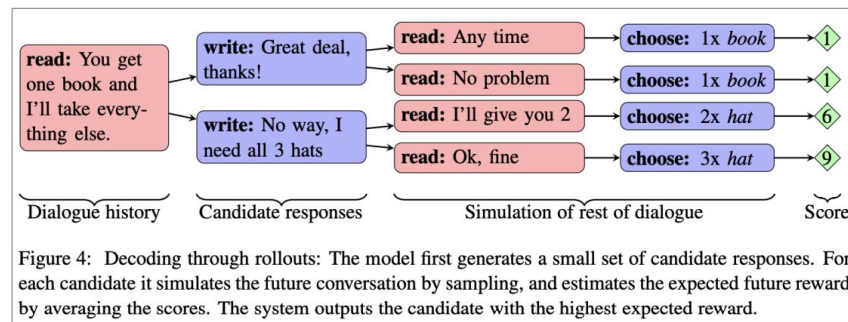
Type Message Here:

Deal or No Deal? End-to-End Learning for Negotiation Dialogues, Lewis et al, 2017 <https://arxiv.org/pdf/1706.05125.pdf>



# Negotiation dialogue

- They find that training seq2seq systems for the standard maximum likelihood (ML) objective yields **fluent** but **strategically poor** dialogue agents
- They also use **Reinforcement Learning** to optimize for a discrete reward (with the agents playing against themselves during training)
- **Potential problem:** if the agents just optimize just the RL goal while playing against each other, they might diverge from English!



\*This observation led to an unfortunate media over-reaction: <https://www.skynettoday.com/briefs/facebook-chatbot-language/>  
Deal or No Deal? End-to-End Learning for Negotiation Dialogues, Lewis et al, 2017 <https://arxiv.org/pdf/1706.05125.pdf>



# Negotiation dialogue

- In 2018, Yarats et al proposed another dialogue model for the negotiation task, that separates the **strategic** aspect from the **NLG** aspect.
  - This separation was standard in “old” discourse/dialog NLG approaches
- Each **utterance**  $x_t$  has a corresponding **discrete latent variable**  $z_t$
- $z_t$  is learned to be a **good predictor of future events** in the dialogue (future messages, ultimate strategic outcome), but not a predictor of  $x_t$  itself
- This means that “ $z_t$  *learns to represent  $x_t$ 's effect on the dialogue, but not the words of  $x_t$* ”
- This approach seems useful for controllability, interpretability, easier to learn strategy, etc.

Hierarchical Text Generation and Planning for Strategic Dialogue, Yarats et al, 2018 <https://arxiv.org/pdf/1712.05846.pdf>



# Generation Tasks and Approaches - Storytelling

(Read this on your own time)



# Storytelling

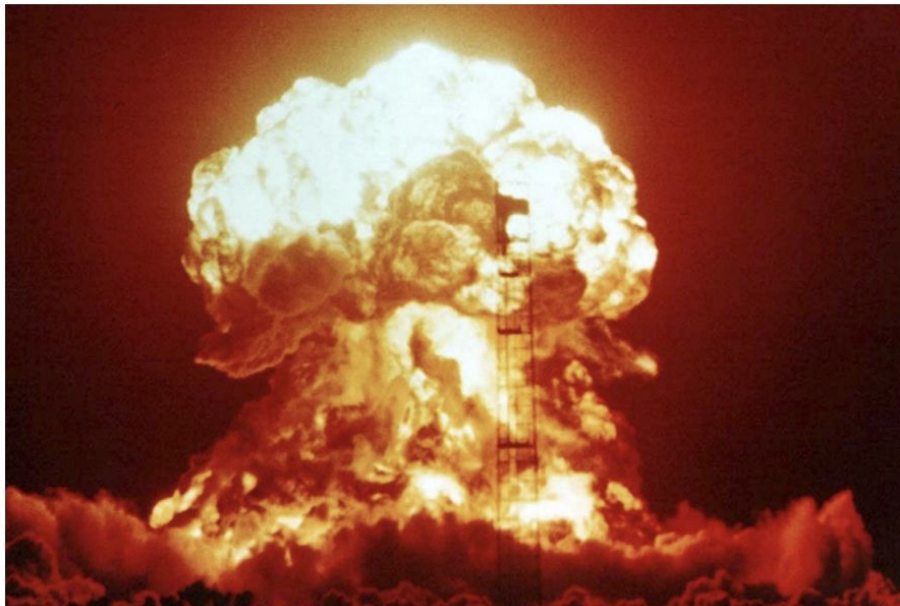
- Most neural storytelling work uses some kind of prompt
  - Generate a story-like paragraph given an image
  - Generate a story given a brief writing prompt
  - Generate the next sentence of a story, given the story so far (story continuation)
    - This is different to the previous two, because we are not concerned with the system's performance over several generated sentences
- Neural storytelling is taking off!
  - The first workshop in 2018
  - It held a competition! (generate a story to accompany a sequence of 5 images)

Storytelling Workshop 2019: <http://www.visionandlanguage.net/workshop2019/>



# Storytelling

This isn't straightforward as supervised image-captioning. There was no paired data to learn from!



**Generated story about image**

Model: Taylor Swift Lyrics

*"You have to be the only light bulb  
in the night sky, I thought, Oh,  
God, it 's so dark out of me that I  
missed you , I promise."*

Generating Stories about Images, <https://medium.com/@samim/generating-stories-about-images-d163ba41e4ed>



# Storytelling

- In 2018, Fan et al. released a new story generation dataset collected from Reddit's WritingPrompts subreddit
- Each story has an associated brief writing prompt

**Prompt:** The Mage, the Warrior, and the Priest

---

**Story:** A light breeze swept the ground, and carried with it still the distant scents of dust and time-worn stone. The Warrior led the way, heaving her mass of armour and muscle over the uneven terrain. She soon crested the last of the low embankments, which still bore the unmistakable fingerprints of haste and fear. She lifted herself up onto the top the rise, and looked out at the scene before her. [...]

---

Hierarchical Neural Story Generation, Fan et al, 2018 <https://arxiv.org/pdf/1805.04833.pdf>





# Storytelling

Fan et al. also proposed a **complex seq2seq prompt-to-story** model:

- It's **convolutional-based**
  - This makes it faster than RNN-based seq2seq
- **Gated multi-head multi-scale self-attention**
  - The self attention is important for capturing long-range context
  - The gates allow the attention mechanism to be more selective
  - The different attention heads attend at different scales - this means there are different attention mechanisms dedicated to retrieving fine-grained information and coarse-grained information
- **Model fusion**
  - Pretrain one seq2seq model, then train a second seq2seq model that has access to the hidden states of the first
  - The idea is that the first seq2seq learns general LM and the second learns to condition on the prompt

Hierarchical Neural Story Generation, Fan et al, 2018 <https://arxiv.org/pdf/1805.04833.pdf>



# Storytelling

The results are impressive!

- **Related** to prompt
- **Diverse**; non-generic
- Stylistically **dramatic**

---

## Example Prompt 1: Aliens start abducting humans

---

It has been two weeks, and the last of my kind has gone. It is only a matter of time until there will be nothing left. I'm not sure what the hell is going on... I can't think. I can hear a distant scream. I think of a strange, alien sound. I try to speak, but am interrupted by something, something that feels like a drum, I ca not tell. I mean I'm just a little bit older than an average human. But I can, and I can feel the vibrations . I hear the sound of a ship approaching. The ground quakes at the force of the impact, and a loud explosion shatters the silence.

- However, mostly atmospheric/descriptive/scene-setting; **less events/plot**
- When generating for longer, mostly **stays on the same idea**, without moving forward to new ideas - coherence issues

Hierarchical Neural Story Generation, Fan et al, 2018 <https://arxiv.org/pdf/1805.04833.pdf>



# Storytelling - GPT2 - transformer-based - even better :-)

|                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYSTEM<br>PROMPT<br>(HUMAN-<br>WRITTEN)                   | <i>In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| MODEL<br>COMPLETION<br>(MACHINE-<br>WRITTEN,<br>10 TRIES) | <p>The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.</p> <p>Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.</p> <p>Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.</p> <p>Pérez and the others then ventured further into the valley. ...</p> |



# Storytelling - challenges

Stories generated by neural LMs can sound fluent...but are **meandering, nonsensical, with no coherent plot. What's missing?**

LMs model **sequence of words**. Stories are **sequence of events**. To tell a story, it's lot more complicated!

- Events and the causality between events (Martin et al. 2018)
- Characters, personalities, motivations, relationships
- State of the world (who, what, where, why) (Yejin Choi research group)
- Narrative structure (e.g., exposition -> conflict -> resolution) (Fan et al. 2019)
- Good storytelling principles (don't introduce a story element then never use it)

Event Representations for Automated Story Generation with Deep Neural Nets, Martin et al, 2018 <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17046/15769>

Yejin Choi research group: <https://homes.cs.washington.edu/~yejin/>

Strategies for Structuring Story Generation, Fan et al, 2019 <https://arxiv.org/pdf/1902.01109.pdf>



# Evaluation



# Automatic evaluation metrics for NLG

## **Word overlap based metrics** (BLEU, ROUGE, METEOR, F1, etc.)

- We know that they're not ideal for machine translation
- They're much worse for summarization, which is more open ended than machine translation
- Unfortunately, ROUGE also typically rewards extractive summarization systems more than abstractive systems
- And they're much, much worse for dialogue, which is even more open-ended than summarization.
  - Similarly for, e.g., story generation, poetry.....(you don't want to imagine...)



# Word overlap metrics are not good for dialogue [Liu et al., ACL 2017]

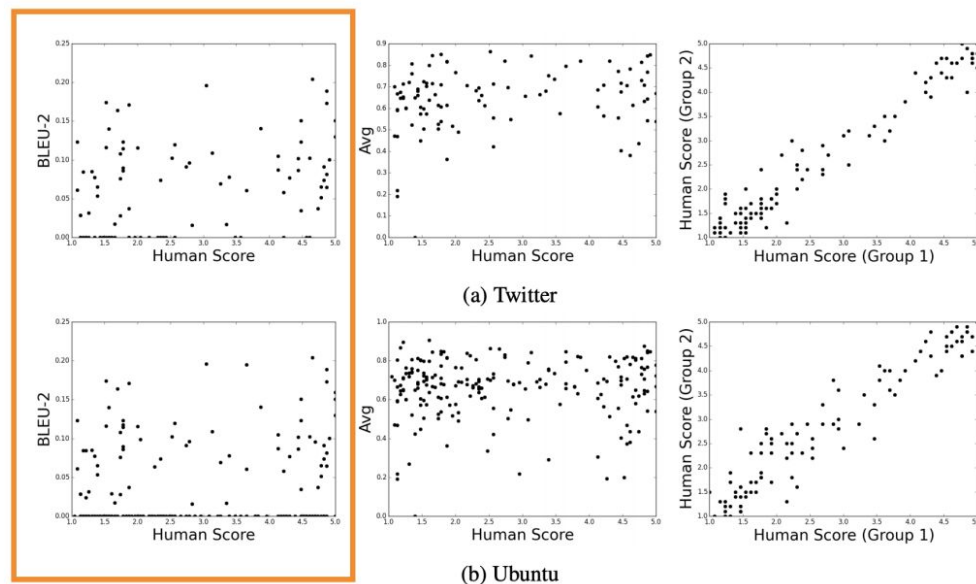
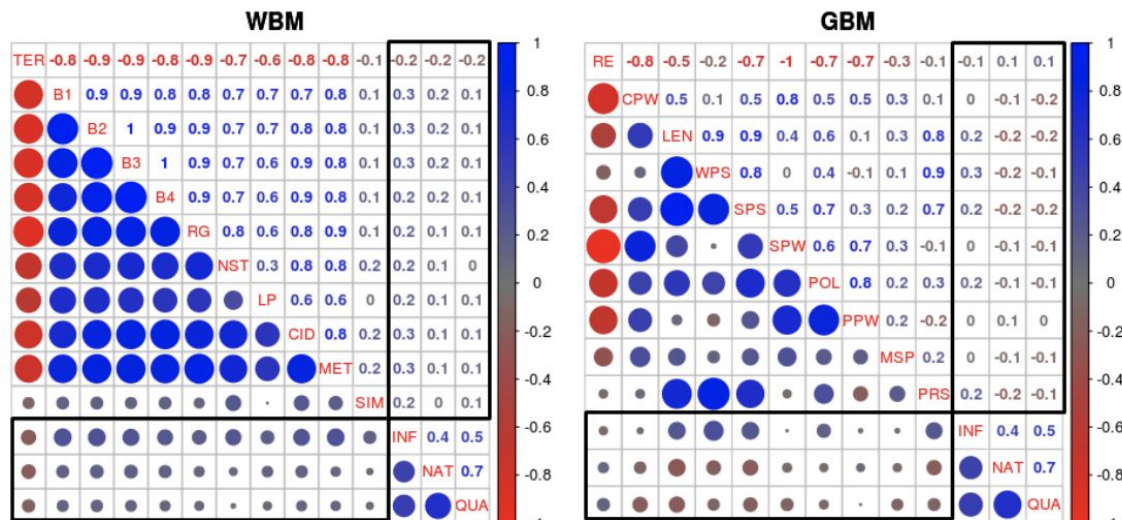


Figure 1: Scatter plots showing the correlation between metrics and human judgements on the Twitter corpus (a) and Ubuntu Dialogue Corpus (b). The plots represent BLEU-2 (left), embedding average (center), and correlation between two randomly selected halves of human respondents (right).

How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation, Liu et al, 2017 <https://arxiv.org/pdf/1603.08023.pdf>



## Other existing metrics are not good either for dialogue [Novikova et al., ACL 2017]



WBM: **Word-based Metrics**  
GBM: **Grammar-based metrics**

Figure 1: Spearman correlation results for TGEN on BAGEL. Bordered area shows correlations between human ratings and automatic metrics, the rest shows correlations among the metrics. Blue colour of circles indicates positive correlation, while red indicates negative correlation. The size of circles denotes the correlation strength.

Why We Need New Evaluation Metrics for NLG, Novikova et al, 2017 <https://arxiv.org/pdf/1707.06875.pdf>



# Automatic evaluation metrics for NLG

- What about **perplexity**?
  - Captures how powerful your LM is, but **doesn't tell you anything about generation** (e.g., if your decoding algorithm is bad, perplexity is unaffected)
- **Word embedding based metrics?**
  - Main idea: compare the **similarity of the word embeddings**, not just the overlap of the words themselves.
  - Unfortunately, still **doesn't correlate well with human judgements** for open-ended tasks like dialogue



# Humans are inconsistent! [See et al., NAACL 2019]

- What if we got access to thousands of humans?
- **Conducting human evaluation is very difficult!**
  - Inconsistent
  - Illogical
  - Lose concentration
  - Misinterpret your question
  - Can't always explain why they feel the way they do
- **Solution:** detailed human evaluation system that separates out important factors that contribute to overall quality, **but still very difficult according to See et al. 2019!**

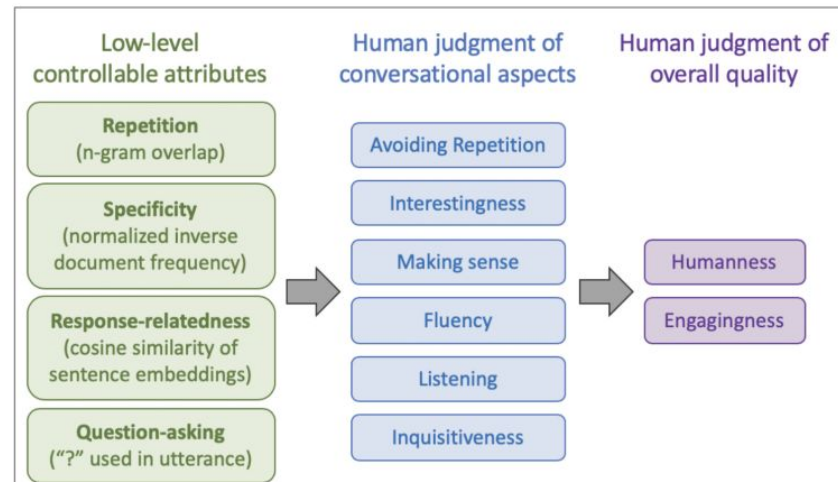


Figure 1: We manipulate four low-level attributes and measure their effect on human judgments of individual conversational aspects, as well as overall quality.

What makes a good conversation? How controllable attributes affect human judgments, See et al, 2019 <https://arxiv.org/pdf/1902.08654.pdf>



# Automatic evaluation metrics for NLG

Thus, bad news....no automatic metrics to adequately capture overall quality

But we can define **more focused automatic metrics** to capture particular aspects of generated text:

- Fluency (compute probability w.r.t. Well-trained LM)
- Correct style (prob w.r.t LM trained on target corpus)
- Diversity (rare word usage, uniqueness of n-grams)
- Relevance to input (semantic similarity measures)
- Simple things like length and repetition
- Task-specific metrics, e.g., compression rate for summarization



# Possible new ways for NLG eval?

- **Corpus-level metrics**
  - Should an eval metric be applied to each example in the test set independently, or a function of the whole corpus?
  - e.g. if a dialogue model always gives the same generic answer to every example in the test set, it should be penalized
- Eval metrics that measure the **diversity-safety tradeoff**
- Human eval for free
  - **Gamification**: make the task (e.g. talking to a chatbot) fun, so humans provide supervision and implicit evaluation for free
- **Adversarial discriminator** as an evaluation metric
  - Test whether the NLG system can fool a discriminator which is trained to distinguish human text from artificially generated text



# Word embedding metrics are not good for dialogue

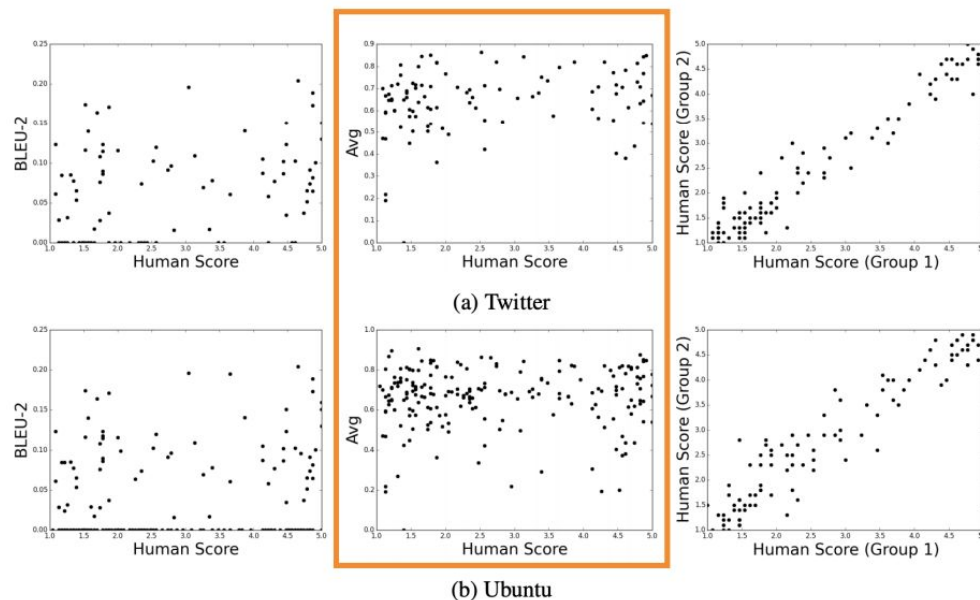


Figure 1: Scatter plots showing the correlation between metrics and human judgements on the Twitter corpus (a) and Ubuntu Dialogue Corpus (b). The plots represent BLEU-2 (left), embedding average (center), and correlation between two randomly selected halves of human respondents (right).

How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation, Liu et al, 2017 <https://arxiv.org/pdf/1603.08023.pdf>



# Summary

- Natural language generation **has MUCH to do!**
  - Bottleneck on generating **natural** responses
- Changing the **loss function** is worthwhile direction to try to optimize for certain properties
  - When it's not possible (not differentiable), use **RL**
- Alternative to maximum likelihood training with **teacher forcing**
- The **lack of effective metrics** that are highly correlated to human judgments (this is the **biggest bottleneck** in this area!)
- Among them...storytelling is probably the hardest! Don't even want to think about poetry (which I have not covered...)
  - Even storytelling seems trivial, the problem about modeling the **world, narratives, emotions can be enlightening to other deep learning problems.....**who knows?

