COMP5046
Natural Language Processing

Lecture 8: Language Model and Natural Language Generation

Semester 1, 2020

School of Computer Science

The University of Sydney, Australia

THE UNIVERSITY OF SYDNEY

Dr Caren Han

Caren.Han@sydney.edu.au

0 Announcements

Thank you for all your hard work!

- **We know the assignment 1 is relatively tough**
- We really appreciate the effort you're putting into this course!

Caren Han STAFF
3 hours ago in Assignments – A1

UNPIN STAR WATCHING 159 VIEWS

Howdy all,

First, I would like to thank all of you guys for hard working in A1 and the NLP course.

I hope you enjoy(?!) working on the assignment 1 now. haha 😍😍
One of our students (Daniel) in NLP requested for extending the due date for A1.
(https://edstem.org/courses/3933/discussion/213413)

I definitely understand the situation and also the COVID-19 issue
so I decided to extend the deadline to 23:59pm 24/04/2020.

Hope this gives some more hours for you guys to wrap up the final version!

Let's push it to the *last* breath together!!

Regards,
Caren

0 LECTURE PLAN

I know your brain is working only for assignment 1 now 🤯

so please sit back and enjoy the easy-going lecture today!

Prepare your popcorn and some drink 🍿🥤

0 LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

1. **Language Model**
2. Traditional Language Model
3. Neural Language Model
4. Natural Language Generation
5. NLG Tasks
6. Language Model and NLG Evaluation

What is Language Model

- is the task of predicting what word comes next based on the given words.
- is a probabilistic model which predicts the probability that a sequence of tokens belongs to a language.

Can you come ________

here
there ***close***
hear
their
further
…

$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(t)}$ x^1, x^2, x^3

Given a sequence of words, ***Can, you, come,*** compute the probability distribution of the next word.

$\boldsymbol{x}^{(t+1)}$ *(can be any word in the vocabulary)*

$$P(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})$$

1 Language Model

THE UNIVERSITY OF SYDNEY

What is Language Model

- is the task of predicting what word comes next based on the given words.
- is a probabilistic model which predicts the probability that a sequence of tokens belongs to a language.

Can you come ***here***

Can you come ***there***

here
there
close
hear
their
further
…

P(Can, you, come, here) *vs* *P(Can, you, come, there)*

P(here|can, you, come) ***vs*** ***P(there|can, you, come)***

$$P(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})$$

What is Language Model

- is the task of predicting what word comes next based on the given words.
- is a probabilistic model which predicts the probability that a sequence of tokens belongs to a language.

Can you come **here**

Can you come **there**

here
there
close
hear
their
further
...

P(Can, you, come, here) vs *P(Can, you, come, there)*

P(here|can, you, come)* vs *P(there|can, you, come)

$$P(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(T)}) = P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)}|\,\boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)}|\,\boldsymbol{x}^{(T-1)},\ldots,\boldsymbol{x}^{(1)})$$
$$= \prod_{t=1}^{T} P(\boldsymbol{x}^{(t)}|\,\boldsymbol{x}^{(t-1)},\ldots,\boldsymbol{x}^{(1)})$$

Conditional Probability

1 Language Model

Language Modeling in NLP

- The probabilities returned by a language model are mostly useful to compare the likelihood that different sentences are "good sentences". Useful in **many practical tasks**, for example:

Spell correction/Automatic Speech Recognition

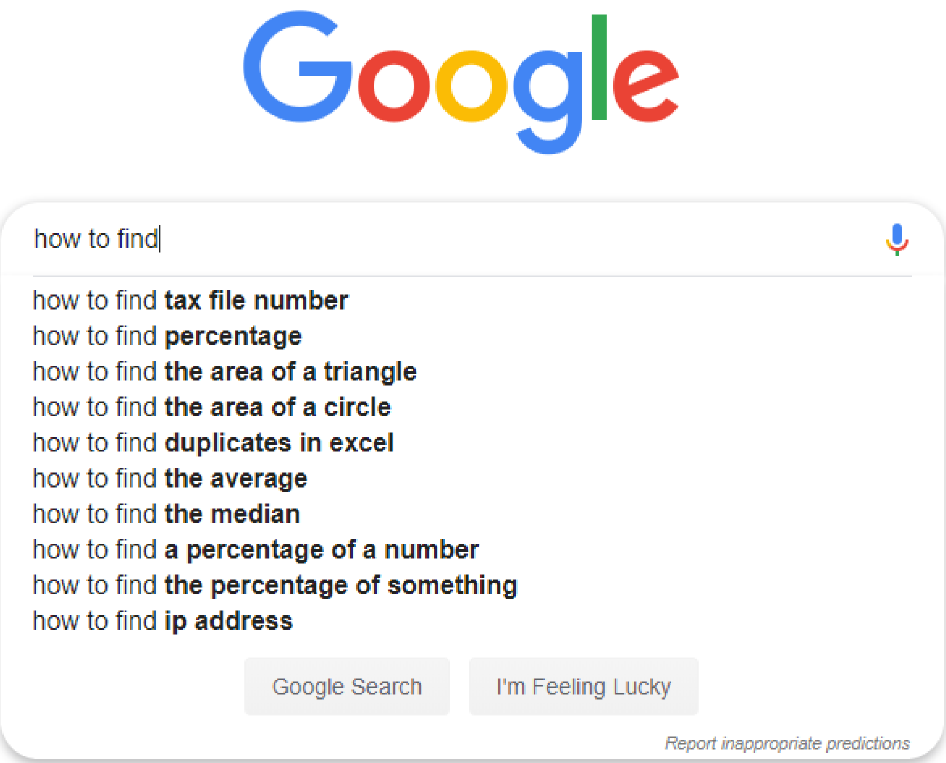
- I would like to read that ***book*** *Closest words= [book, boog, boat, …]*

Natural Language Generation

- Dialogue (chit chat and task-based)
- Abstractive Summarisation
- Machine Translation
- Creative Writing: Story Telling, …

1 Language Model

Do we use Language Model?

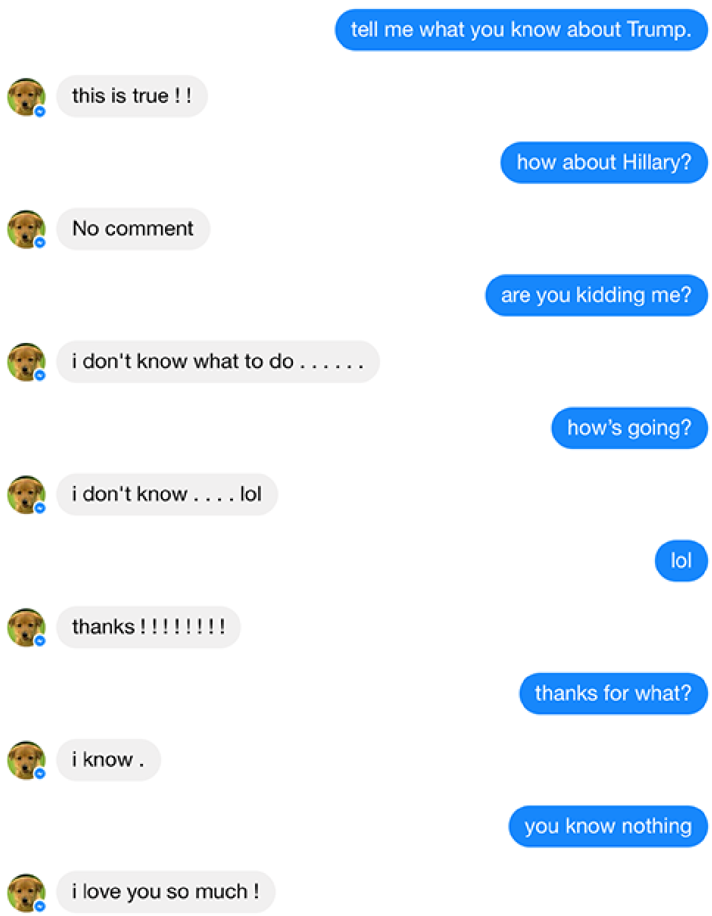


1 Language Model

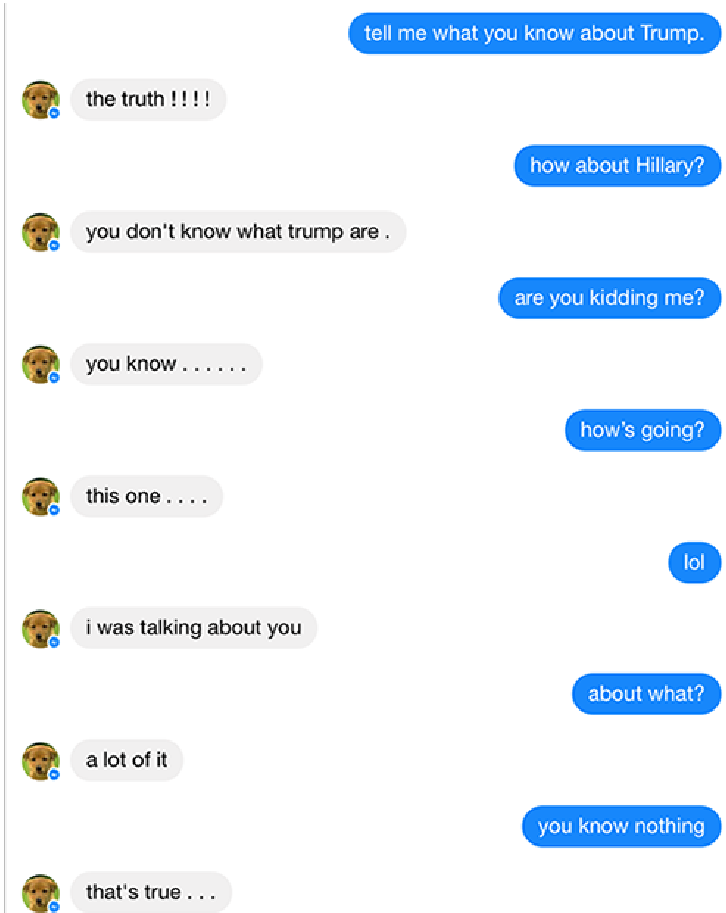
Yes but sometimes fail..

THE UNIVERSITY OF SYDNEY

Language Model in Dialog System



Without anti-language model



With anti-language model

THE UNIVERSITY OF SYDNEY

Language Modeling in Natural Language Generation

Conditional Language Modeling: the task of predicting the next word, given the words so far, and also some other input ***x***:

Natural Language Generation Tasks

- Dialogue (chit chat and task-based)
  x=dialogue history, y=next utterance
- Abstractive Summarisation
  x=input text, y=summarized text
- Machine Translation (in later week)
  x=source sentence, y=target sentence

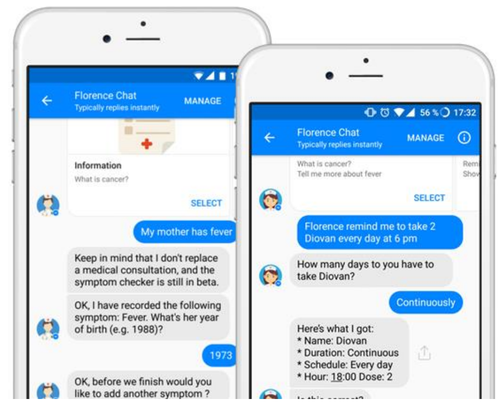
THE UNIVERSITY OF SYDNEY

1 Language Models

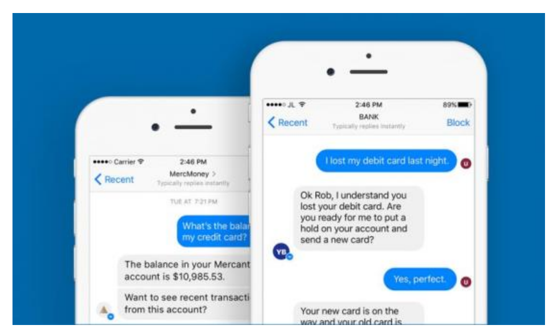
Tips for using Language Model (you already knew!)

It is extremely important to collect and learn the model with the corpus that includes documents about the domain that your system/application will be used.

Medical Documents



Financial Documents



0

LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

1. Language Model
2. **Traditional Language Model**
3. Neural Language Model
4. Natural Language Generation
5. NLG Tasks
6. Language Model and NLG Evaluation

Statistical Language Model (SLM)

- ***Conditional Language Modeling***: the task of predicting the next word, given the words so far, and also some other input ***x***:

$$P(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}) = P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} |\ \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} |\ \boldsymbol{x}^{(T-1)}, \dots, \boldsymbol{x}^{(1)})$$
$$= \prod_{t=1}^{T} P(\boldsymbol{x}^{(t)} |\ \boldsymbol{x}^{(t-1)}, \dots, \boldsymbol{x}^{(1)})$$

"An adorable little boy is spreading smiles"

P(An, adorable, little, boy, is, spreading, smiles)
= P(An) × P(adorable|An) × P(little|An adorable) × P(boy|An adorable little) × P(is|An adorable little boy) × P(spreading|An adorable little boy is) × P(smiles|An adorable little boy is spreading)

Statistical Language Model (SLM)

- ***Conditional Language Modeling***: the task of predicting the next word, given the words so far, and also some other input ***x***:

$$P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) = P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} | \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} | \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)})$$

$$= \prod_{t=1}^{T} P(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})$$

"An adorable little boy is"

*P(**is**|An adorable little boy)?*

Trained Corpus

An adorable little boy is ....
..
.... An adorable little boy is
..
..
An adorable little boy laughed
..
..
..

Simplest method

= Count(An adorable little boy is)/Count(An adorable little boy)
= 30/100 =0.3

Q: What if there is no 'An adorable little boy is' phrase in the corpus?

N-gram Language Models

- *An N-gram is a sequence of N words.*
- *An N-gram model predicts the probability of a given N-gram within any sequence of words in the language.*

"An adorable little boy is spreading smiles"

A ***n*-gram** is a chunk of *n* consecutive words.

- **uni**grams : an, adorable, little, boy, is, spreading, smiles
- **bi**grams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles
- **tri**grams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles
- **4-**grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

2 Traditional Language Models

N-gram Language Models: Exercise

- *Assume that we learn a **tri**gram language model*

“~~An adorable little boy~~ is spreading ___?___”

n-1 words only

(3-1) words only

P(w|is spreading) =
Count(is spreading w) / Count(is spreading)

Trained Corpus

boy is spreading smile……………….
……………………………………………….
………….. boy is spreading rumours
……………………………………………….
……………………………………………….
An adorable little boy is spreading
……………………………………………….
……………………………………………….
……………………………………………….
……………………………………………….
……………………………………………….

P(rumours|is spreading)
= Count(is spreading rumours)/Count(is spreading)
= 500/1000 =0.5

P(smiles|is spreading)
= Count(is spreading smiles)/Count(is spreading)
= 200/1000 =0.2

N-gram Language Models: Beautiful Formula ☺

- ***Simplifying assumption***: *the next word,* $\boldsymbol{x}^{(t+1)}$ *, depends only on the* ***preceding n-1 words****.*

$$P(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(1)}) = P(\boldsymbol{x}^{(t+1)}|\underbrace{\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)}})$$

- *How do we get these n-gram and (n-1)-gram probabilities?*

Counting them!

$$\approx \frac{\text{count}(\boldsymbol{x}^{(t+1)},\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}{\text{count}(\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}$$

THE UNIVERSITY OF SYDNEY

N-gram Language Model Limitation: Trade-off Issue

- *Mostly, n=2 works better than n=1 in n-gram language model*
- *We learned a **tri**gram language model*

"~~An adorable little boy~~ is spreading ___?___"

n-1 words only

(3-1) words only

- *Find the optimal **n is important!** (OOV issue or Model Size issue)*

P(w|is spreading) =

Count(is spreading w) / Count(is spreading)

Need to store count for all n-grams that you saw in the corpus.

If you increase n or corpus,

the model size will be increased!

N-gram Language Model Limitation: Zero Count Issue

P(w|is spreading) =
*Count(**is spreading w**) / Count(**is spreading**)*

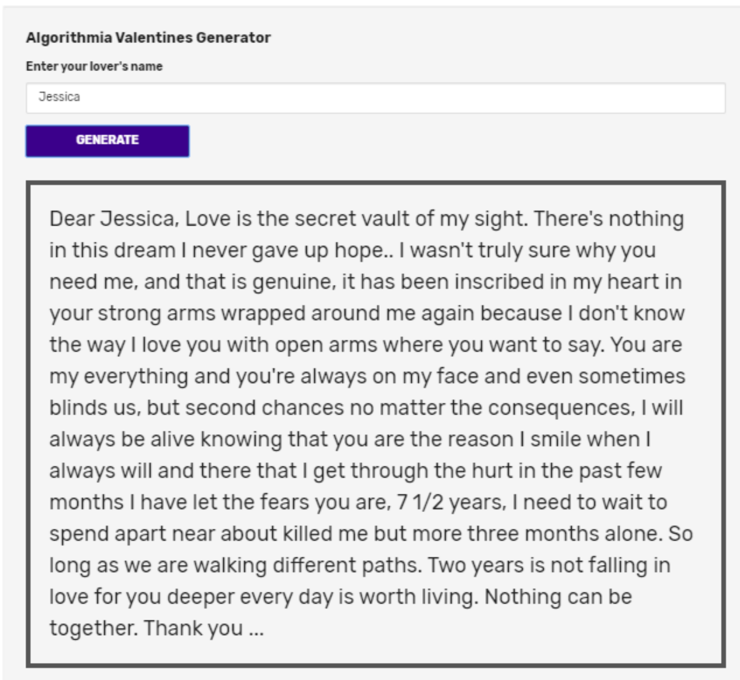
1. *What if the '**is spreading w**' phrase never occurred in the corpus? The probability will be 0.*
   - *Alternative solution: Smoothing (Add small δ to the count for every w in the corpus)*

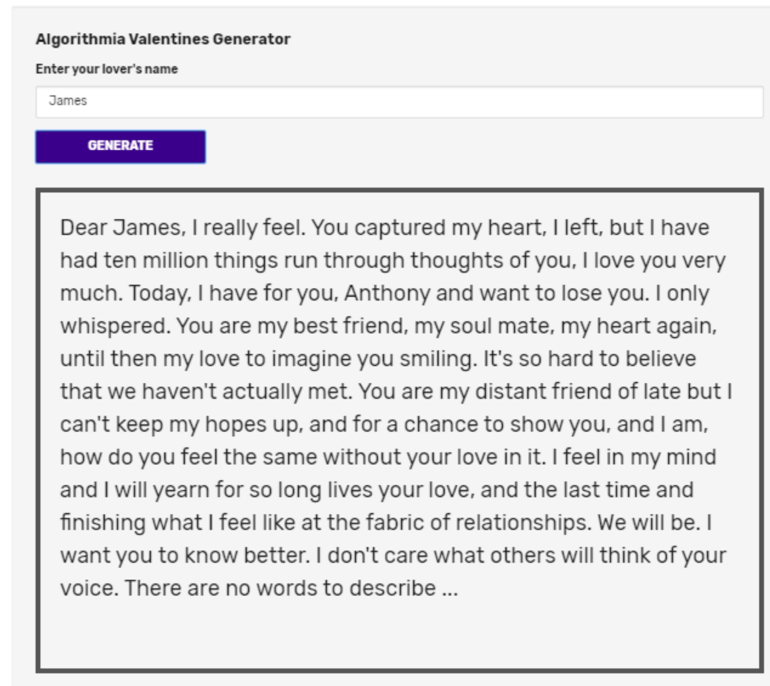
2. *What if the '**is spreading**' phrase never occurred in the corpus? It is impossible to calculate the probability for any **w**.*
   - *Alternative solution: Backoff (Just condition on "spreading" instead)*

THE UNIVERSITY OF SYDNEY

N-gram Language Model Demo

Generating text with a n-gram language model

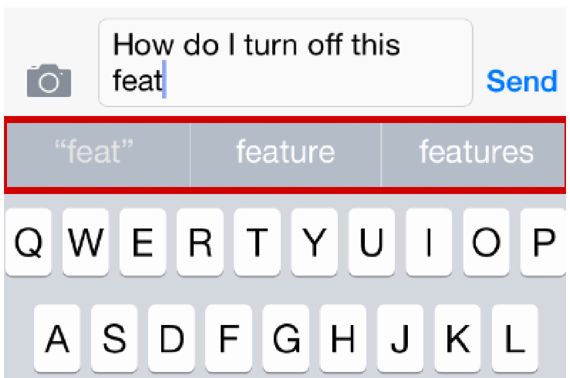


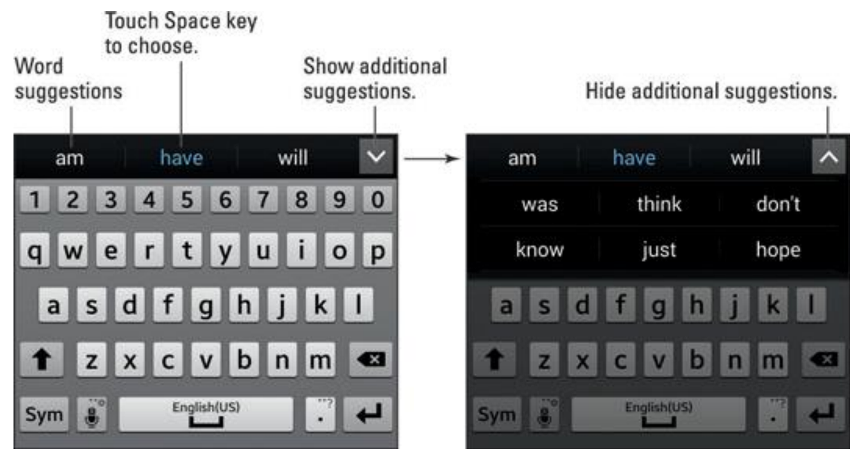


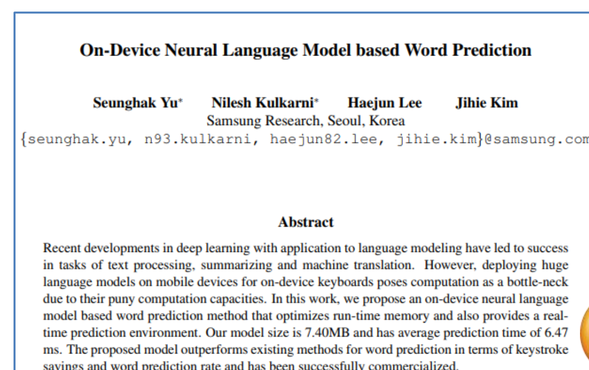
https://algorithmia.com/blog/playing-with-n-grams

Try some Language Model – Word Prediction

Generating text with a n-gram







On-Device Neural Language Model based Word Prediction

Seunghak Yu* Nilesh Kulkarni* Haejun Lee Jihie Kim
Samsung Research, Seoul, Korea
{seunghak.yu, n93.kulkarni, haejun82.lee, jihie.kim}@samsung.com

Abstract

Recent developments in deep learning with application to language modeling have led to success in tasks of text processing, summarizing and machine translation. However, deploying huge language models on mobile devices for on-device keyboards poses computation as a bottle-neck due to their puny computation capacities. In this work, we propose an on-device neural language model based word prediction method that optimizes run-time memory and also provides a real-time prediction environment. Our model size is 7.40MB and has average prediction time of 6.47 ms. The proposed model outperforms existing methods for word prediction in terms of keystroke savings and word prediction rate and has been successfully commercialized.

[(Yu et al. ACL 2018) https://www.aclweb.org/anthology/C18-2028.pdf](https://www.aclweb.org/anthology/C18-2028.pdf)

0 LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

1. Language Model
2. Traditional Language Model
3. **Neural Language Model**
4. Natural Language Generation
5. NLG Tasks
6. Language Model and NLG Evaluation

Traditional Neural Language Model

"~~An adorable little~~ boy is spreading __?__ "

fixed window
window size =3

Pros

No Trade-off issue

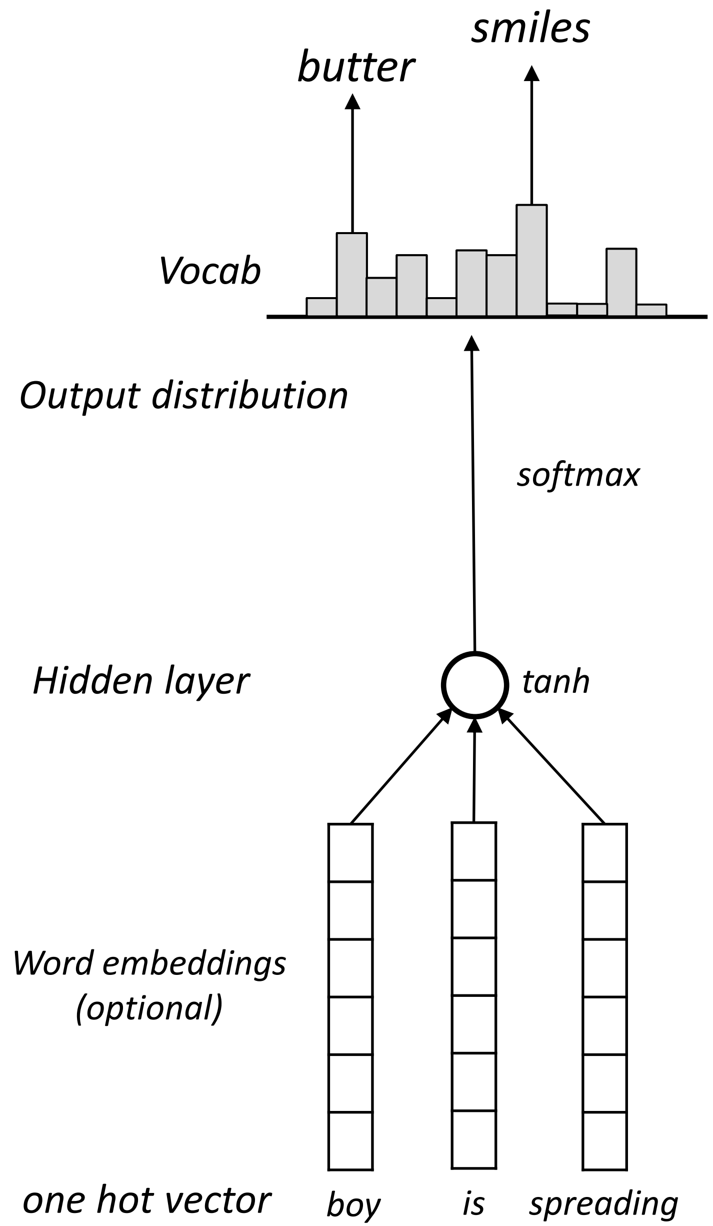
Cons

- ***Window size selection issue***

(increasing window size enlarges W)

- *Input vectors are multiplied by completely different weights in W*

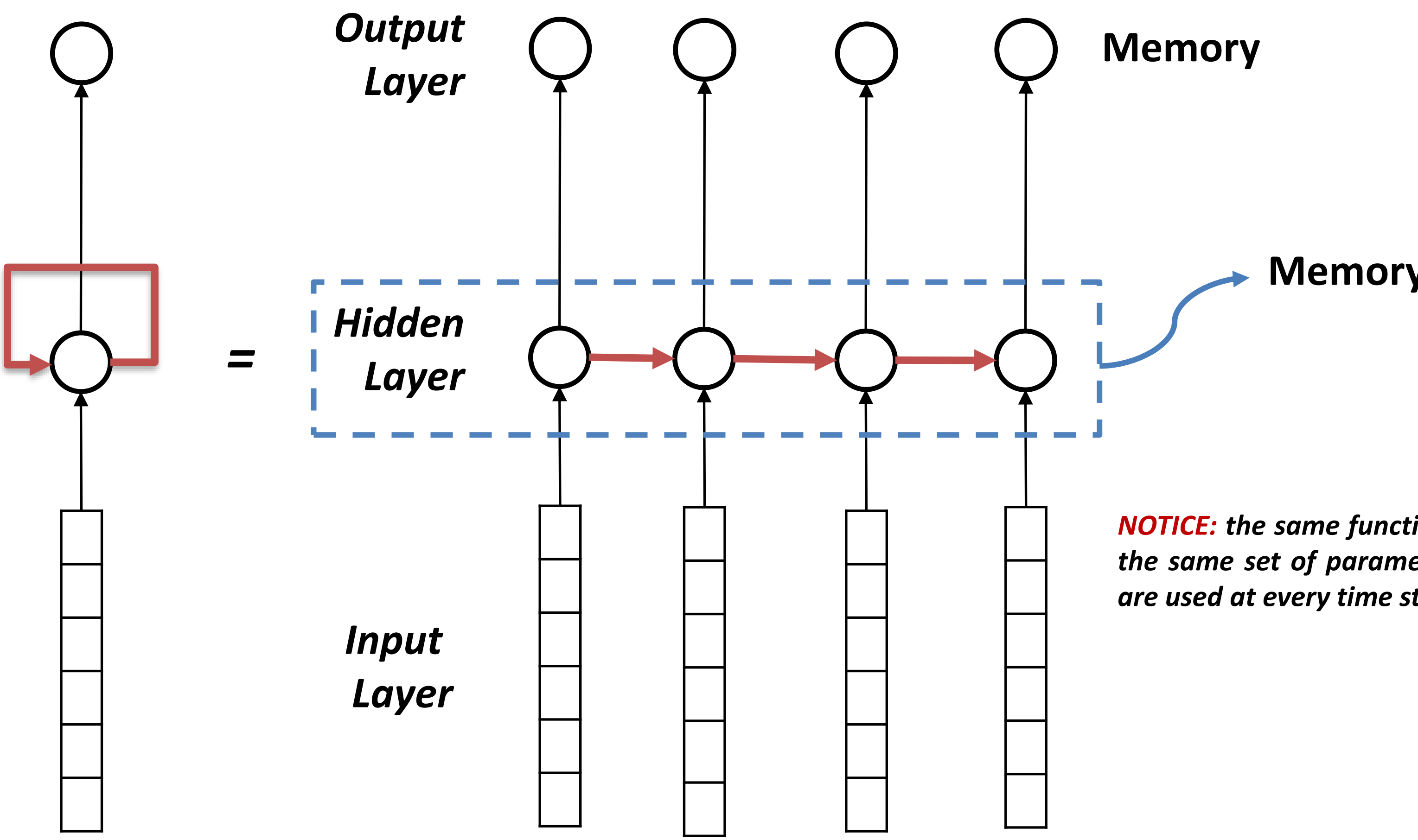
(No symmetry in how the inputs are processed)



3 Neural Language Model

Recap: RNN (Recurrent Neural Network)

Neural Network + Memory = Recurrent Neural Network

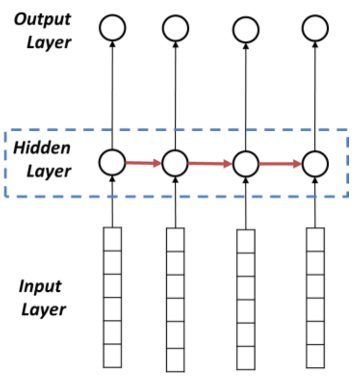


***NOTICE:** the same function and the same set of parameters W are used at every time step*

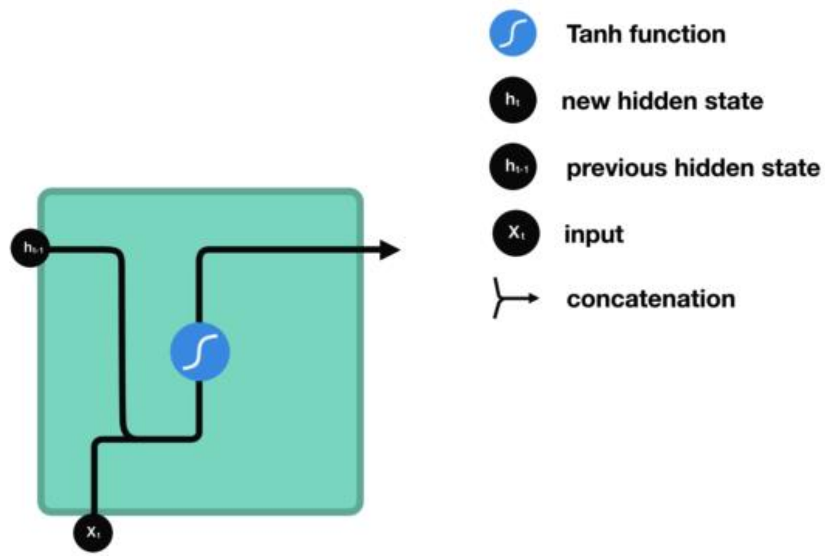
3 Neural Language Model

Recap: RNN (Recurrent Neural Network)

Neural Network + Memory = Recurrent Neural Network



NOTICE: ***the same function and the same set of parameters W are used at every time step***



Previous state *input*

$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

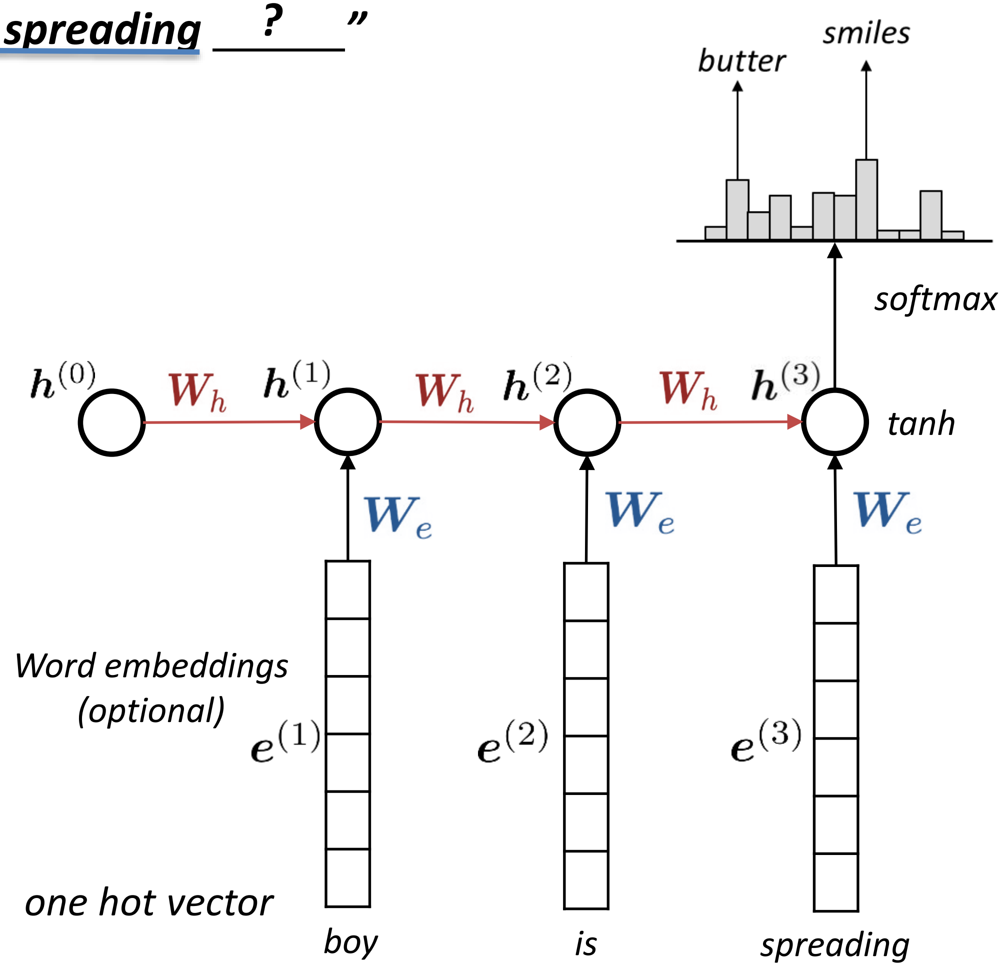
New hidden state

A function with parameters W

RNN-based Language Model

"~~An adorable little~~ boy is spreading ___?___ "

$h^t = tanh(W_h h^{t-1} + W_e e^t + b_1)$



3 Neural Language Model

RNN-based Language Model

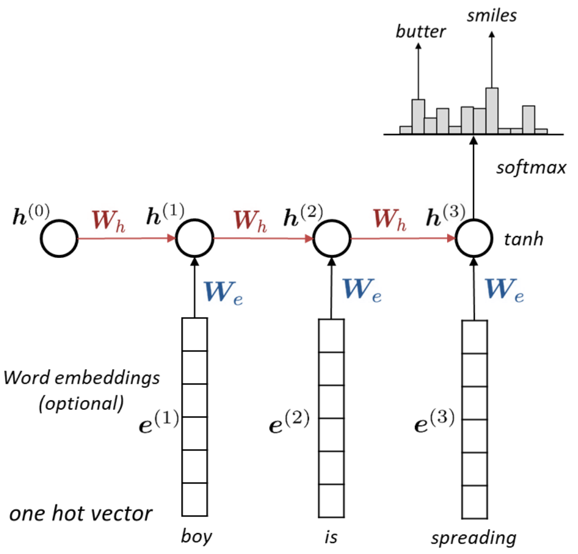
"~~An adorable little~~ boy is spreading __?__"

Pros

- *Can process any length input*
- *Can use information from many step back*
- *Model size does not increase*
- *Same weights applied on every time step (Symmetry)*

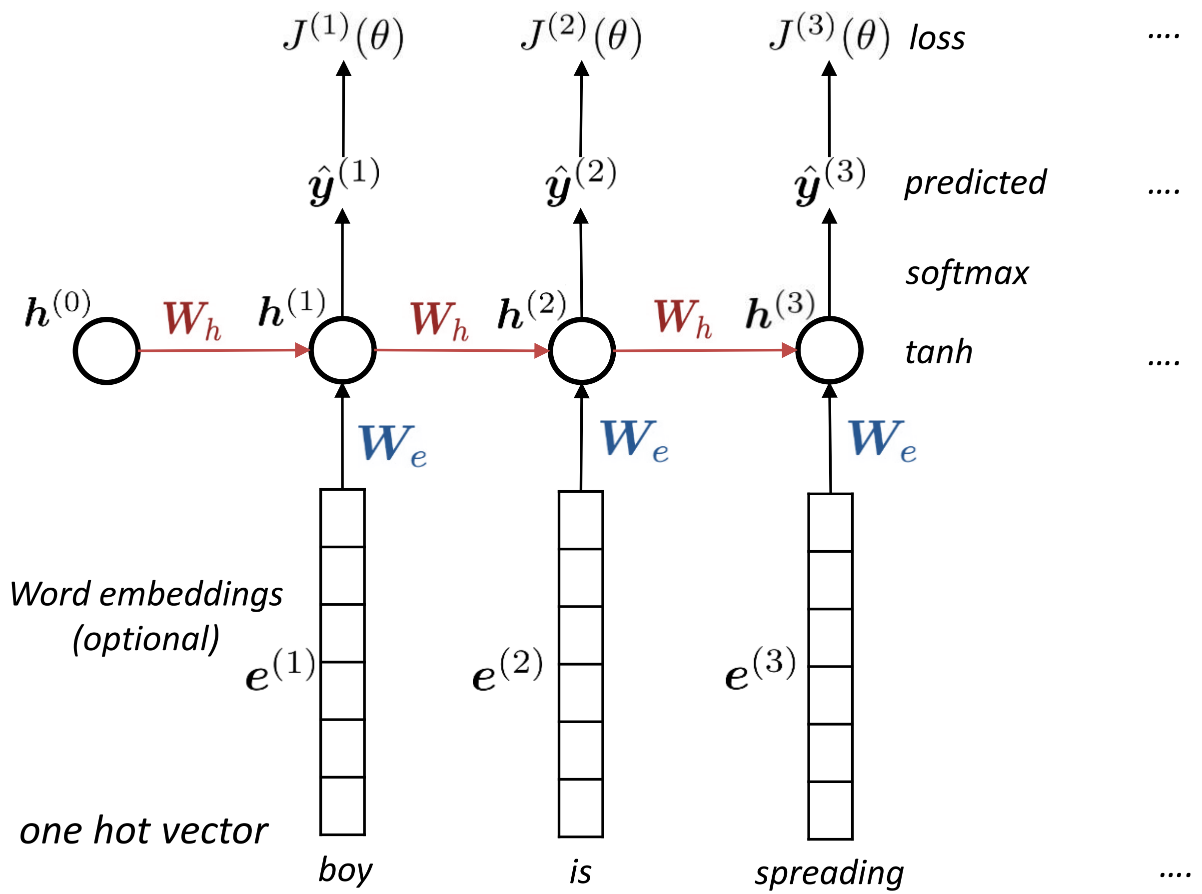
Cons

- *Slow computation*
- *Difficult to access information from many step back (remember what we learned in lecture 4?)*



Training a RNN-based Language Model

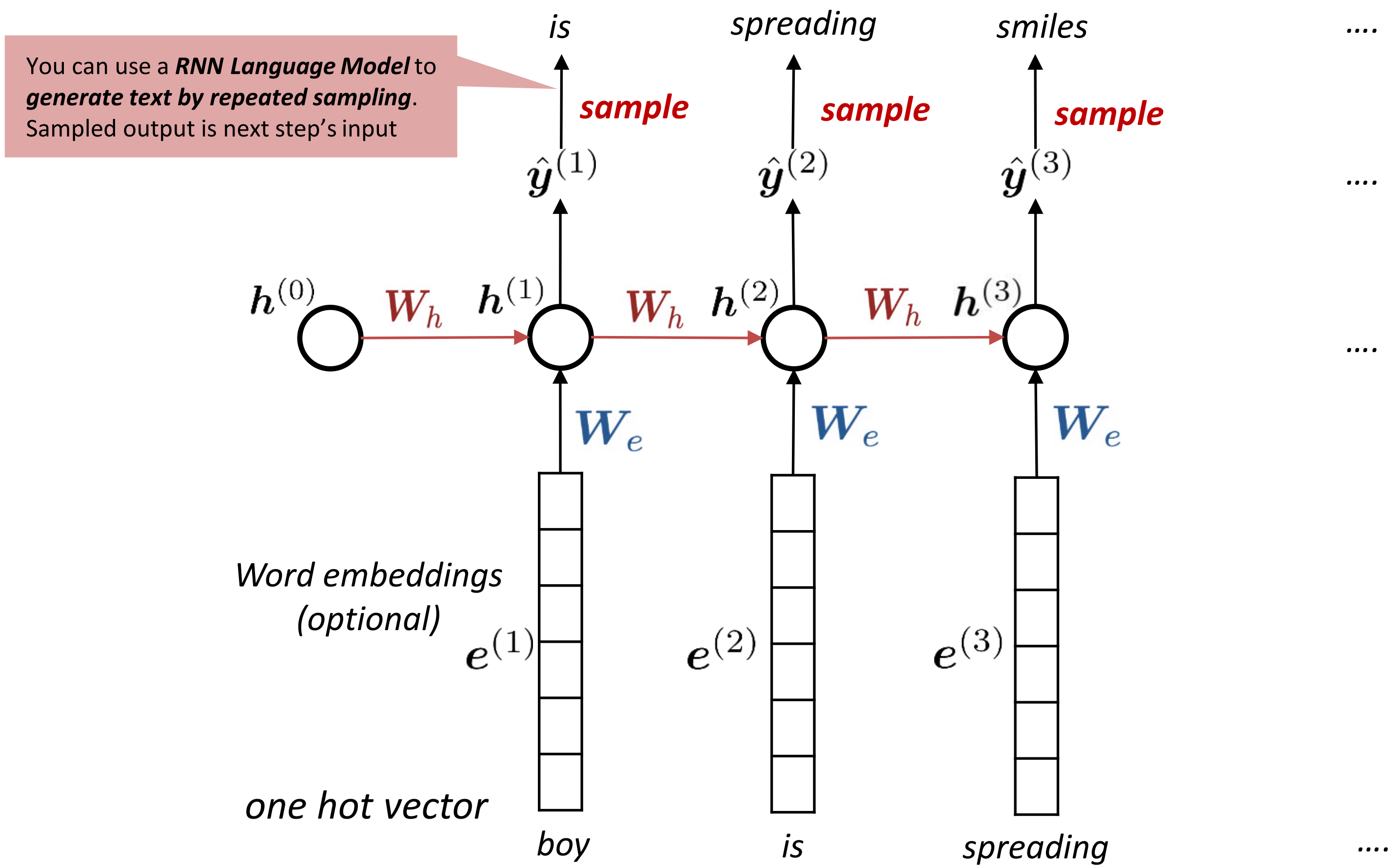
"~~An adorable little~~ boy is spreading __?__"



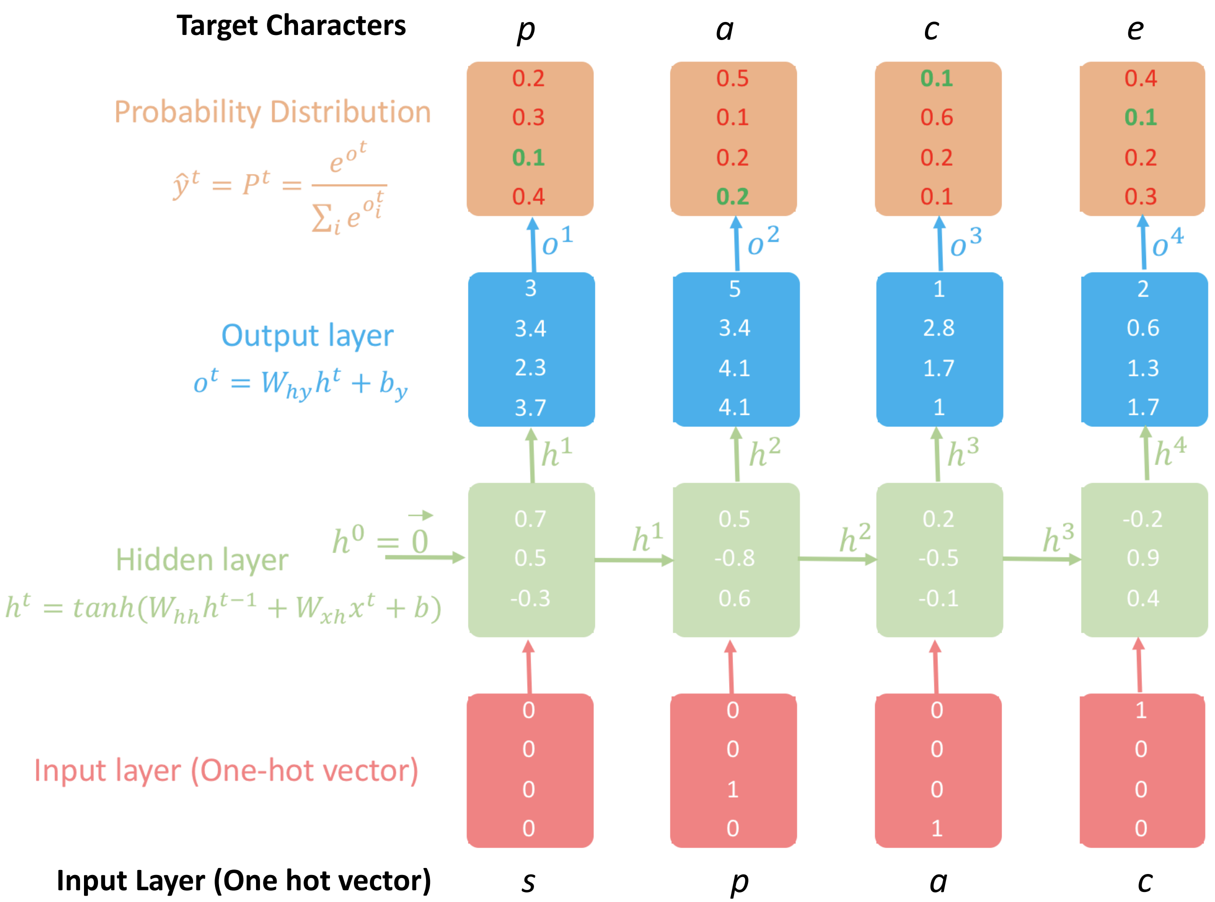
3 Neural Language Model

RNN-based Language Model

"~~An adorable little~~ boy is spreading ___?___"



Recap: Character-based RNN Language Model

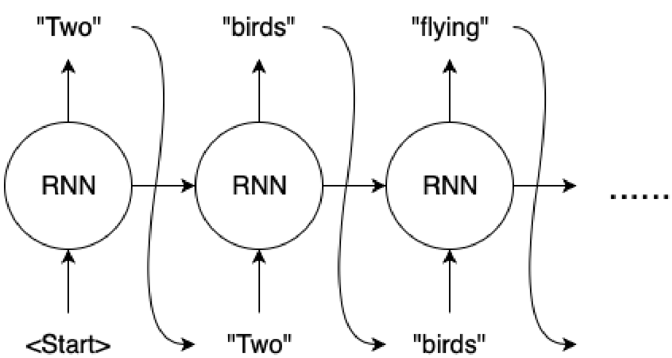


3 Neural Language Model

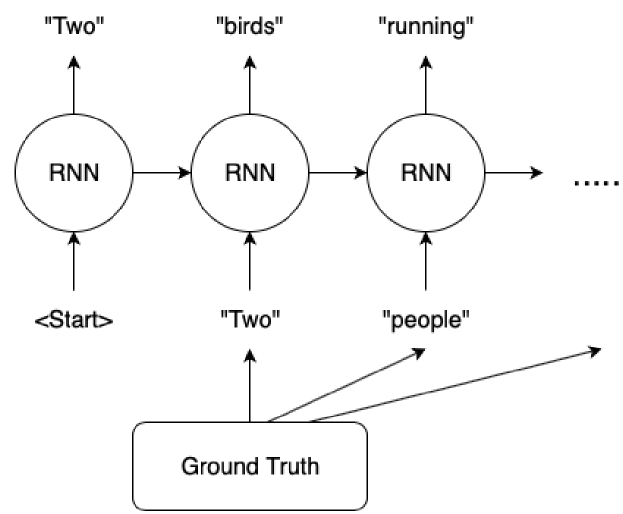
RNN with trained language model

During training, we feed the gold (aka reference) target, regardless of what each cell predicts. This training method is called <u>Teacher Forcing</u>.

Without Teacher Forcing



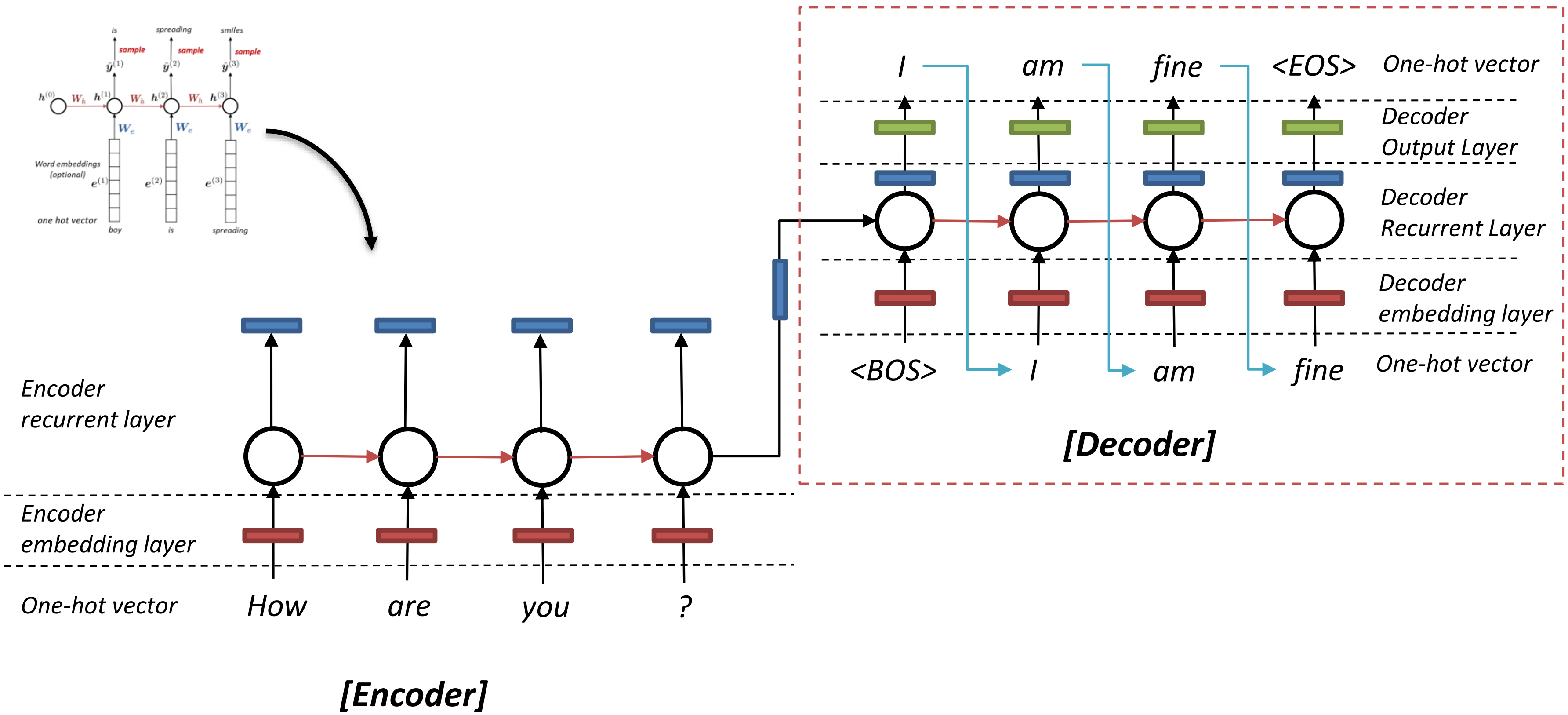
With Teacher Forcing



3 Neural Language Model

Seq2Seq Model with trained language model

During training, we feed the gold (aka reference) target sentence into the decoder, regardless of what the decoder predicts. This training method is called Teacher Forcing.



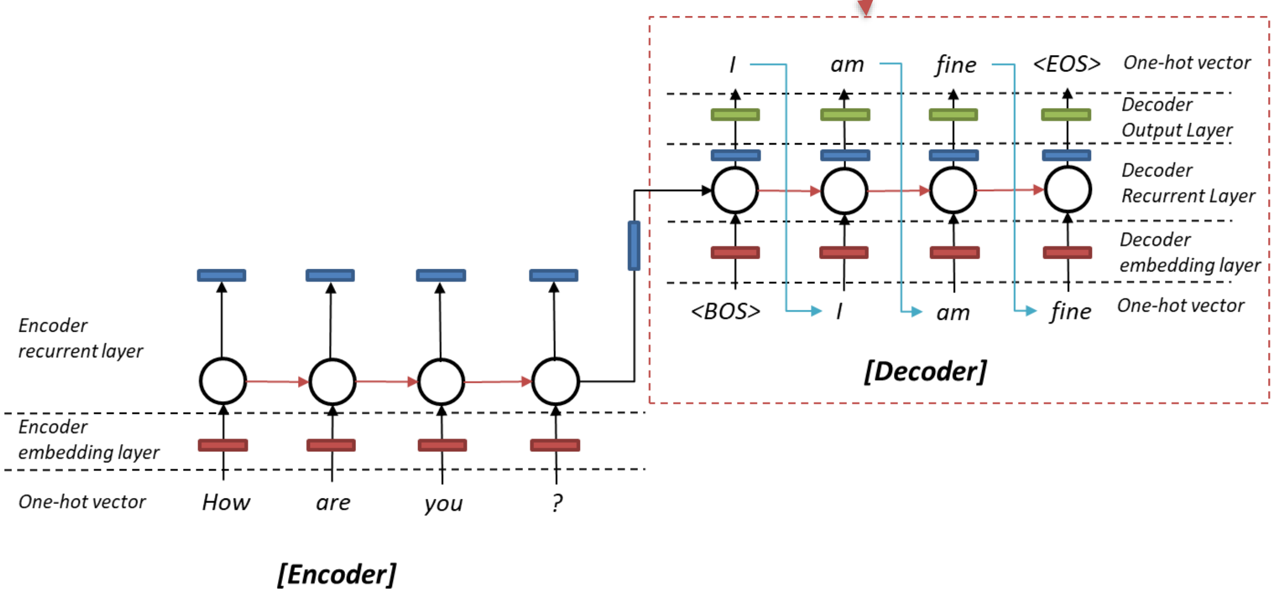
0 LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

1. Language Model
2. Traditional Language Model
3. Neural Language Model
4. **Natural Language Generation**
5. NLG Tasks
6. Language Model and NLG Evaluation

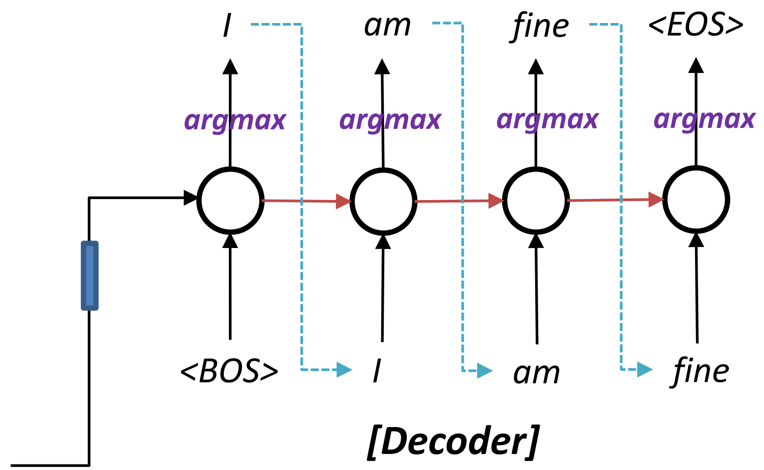
Decoding Algorithm

Now we have trained the conditional neural language model!
How do we use the language model to generate text?
1) Greedy Decoding or 2) Beam Search



Decoding Algorithm 1: Greedy Decoding

- Generate/decode the sentence by taking ***argmax*** on each step of the decoder
  - Take most probable word on each step
- Use that as the next word, and feed it as input on the next step
- Keep going until you produce *<EOS>*



Issue

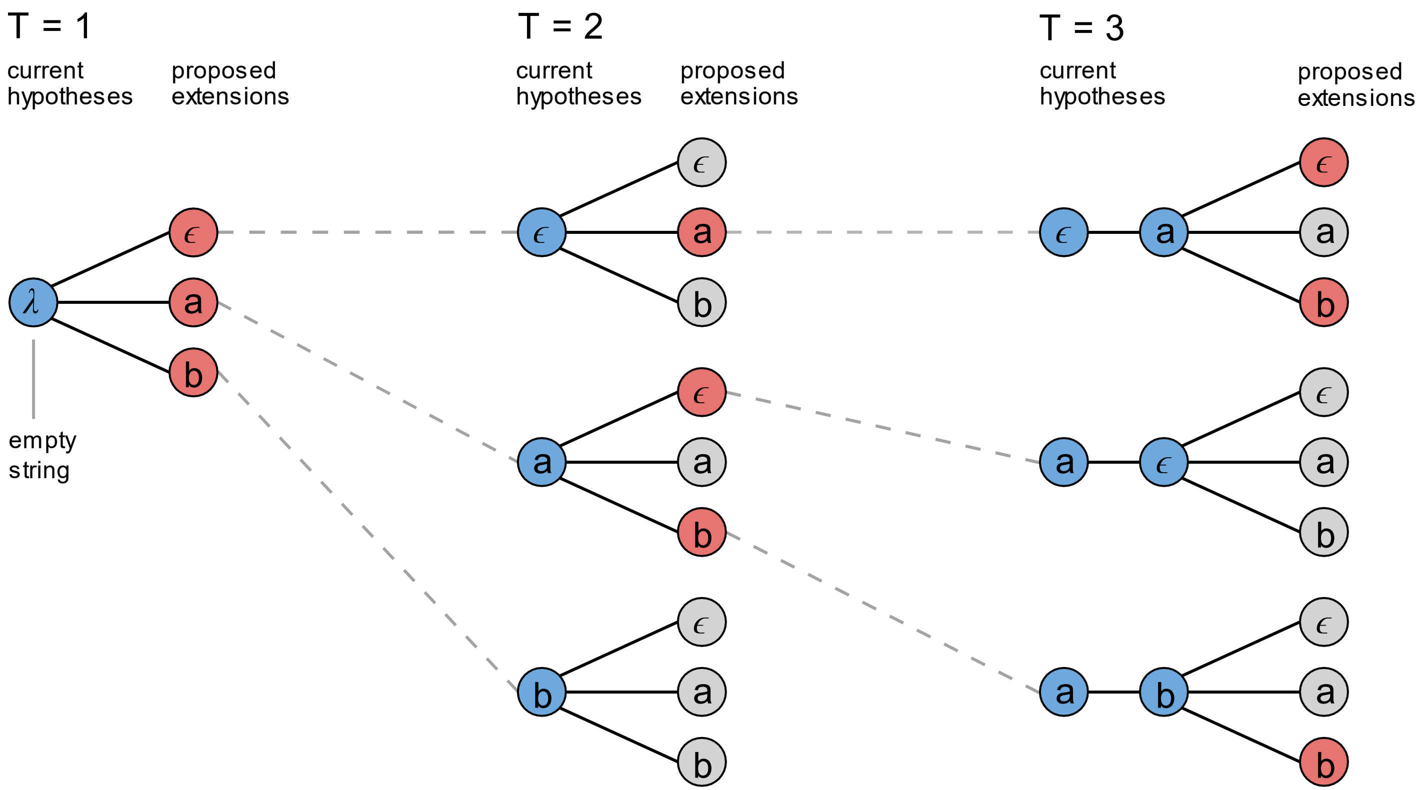
backtracking

- Greedy decoding has no way to undo decisions!! (Ungrammatical, unnatural)
- How to fix this issue?

Exhaustive search decoding: We could try computing all possible sequences

Decoding Algorithm: Beam Search

A standard beam search algorithm with an alphabet of {ϵ,a,b} with a beam size 3.



Decoding Algorithm: Beam Search

- A search algorithm which aims to find a **high-probability sequence** (not necessarily the optimal sequence, though) by tracking multiple possible sequences at once.

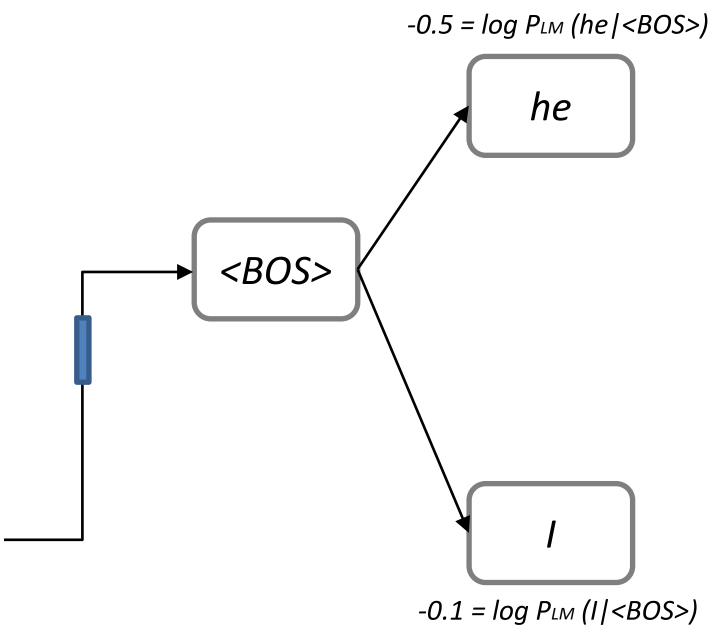
- On each step of decoder, keep track of the ***k most probable*** partial sequences (which we call *hypotheses*)
    - *K is the **beam size** (in practice around 5 to 10)*

- After you reach some stopping criterion, *choose the sequence with the highest probability* (factoring in some adjustment for length)

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Take top k words and compute scores

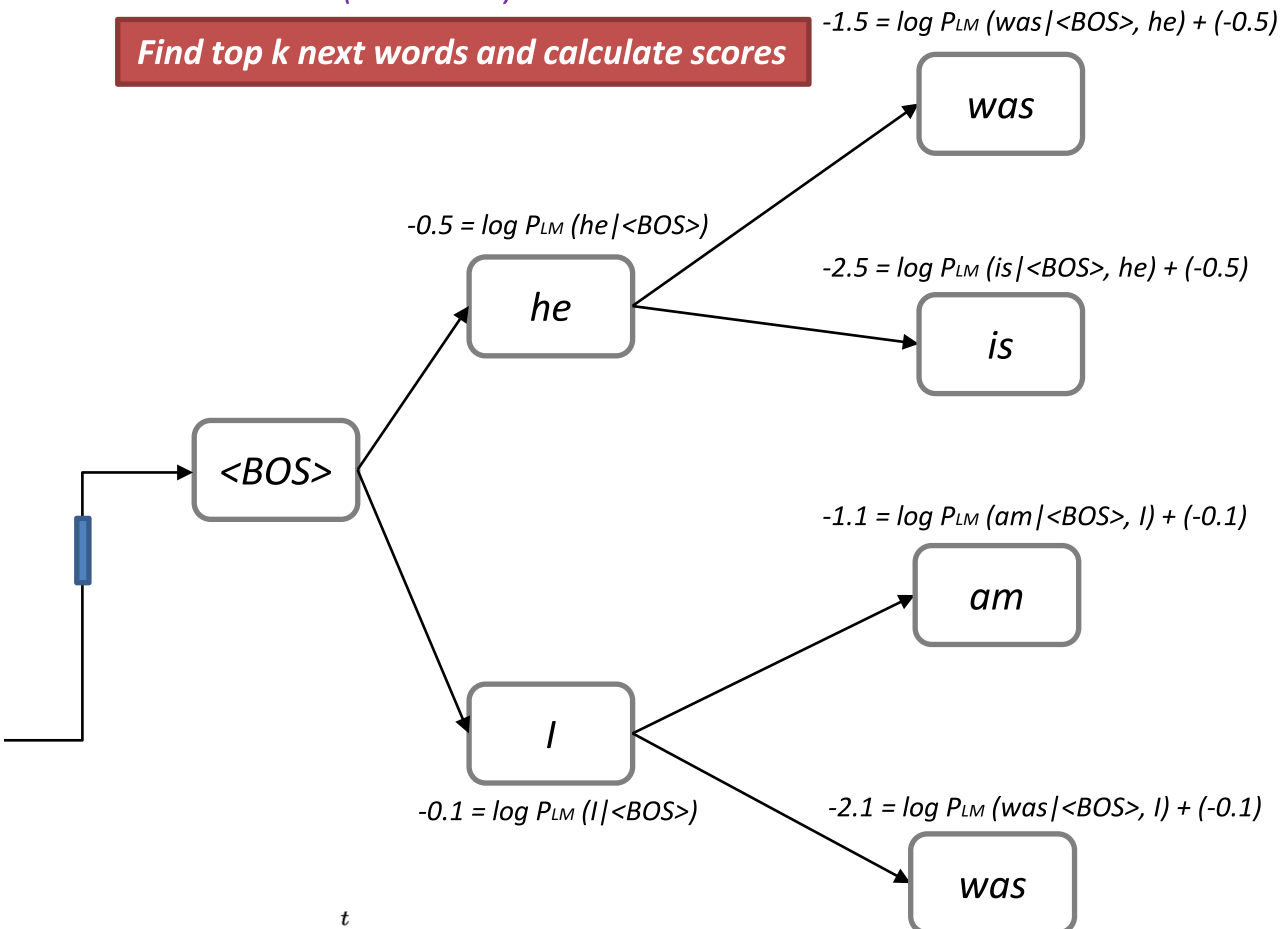


$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Find top k next words and calculate scores

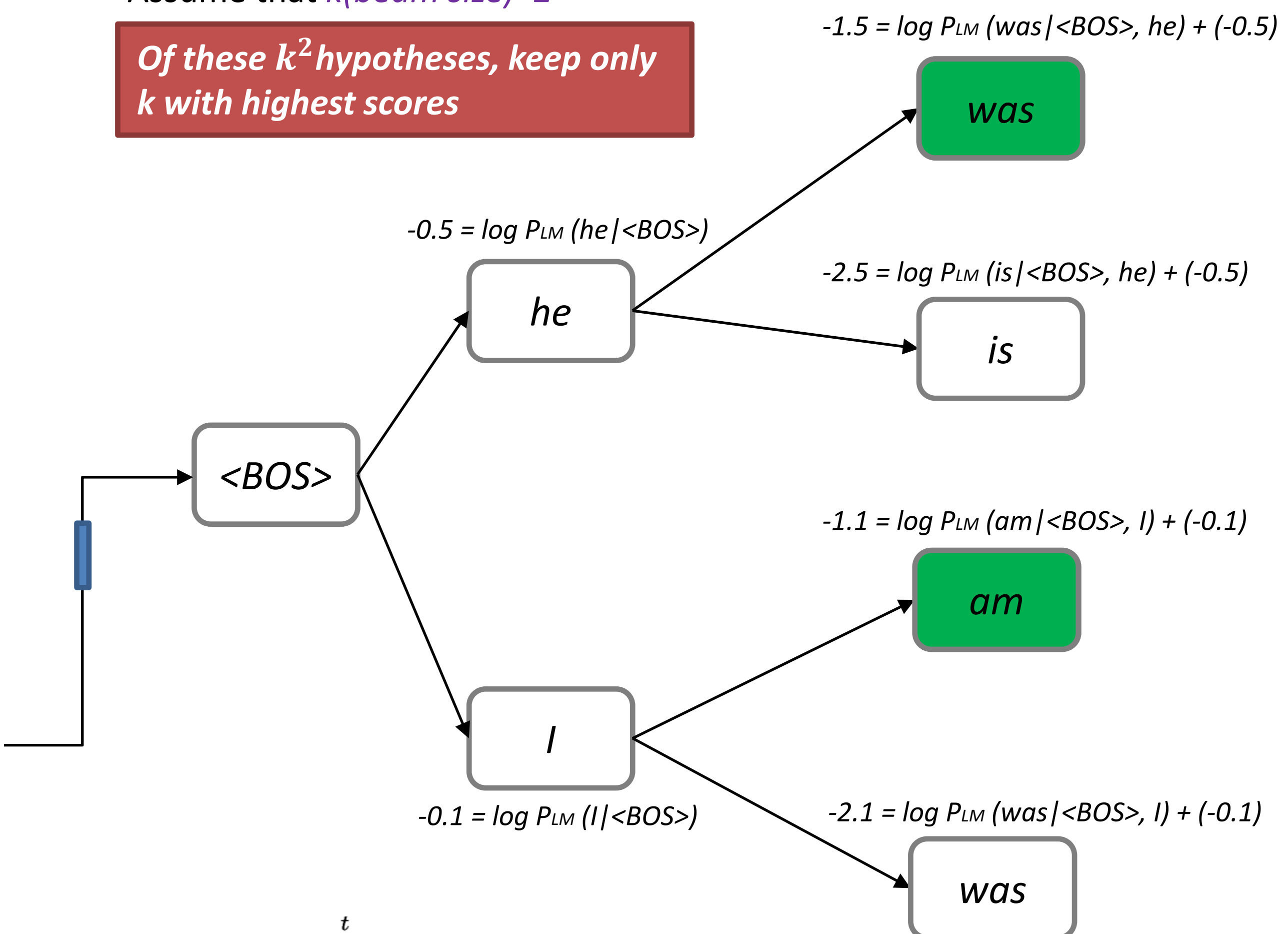


$$\mathrm{score}(y_1,\ldots,y_t)=\sum_{i=1}^{t}\log P_{\mathrm{LM}}(y_i|y_1,\ldots,y_{i-1},x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Of these k^2 hypotheses, keep only k with highest scores

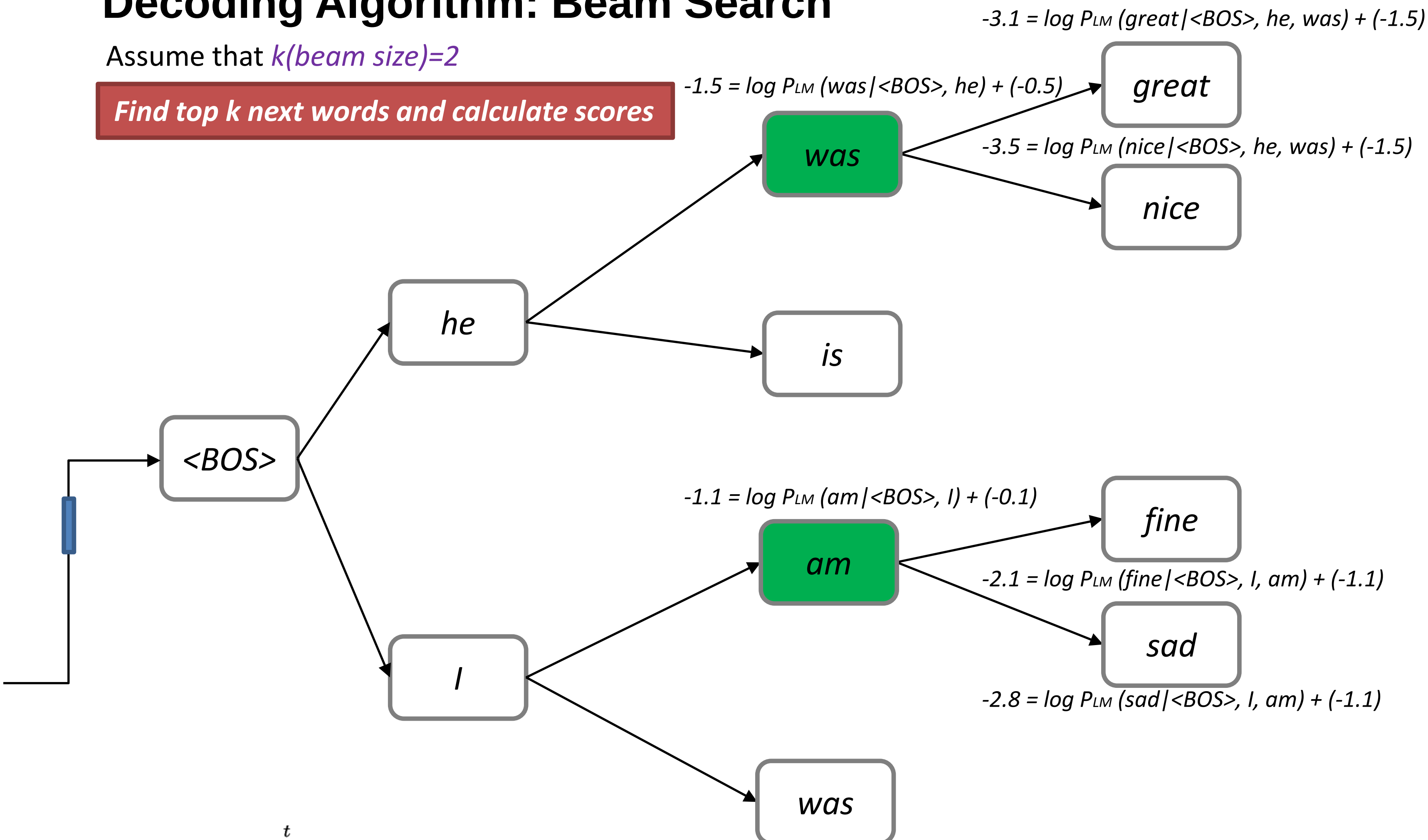


$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Find top k next words and calculate scores

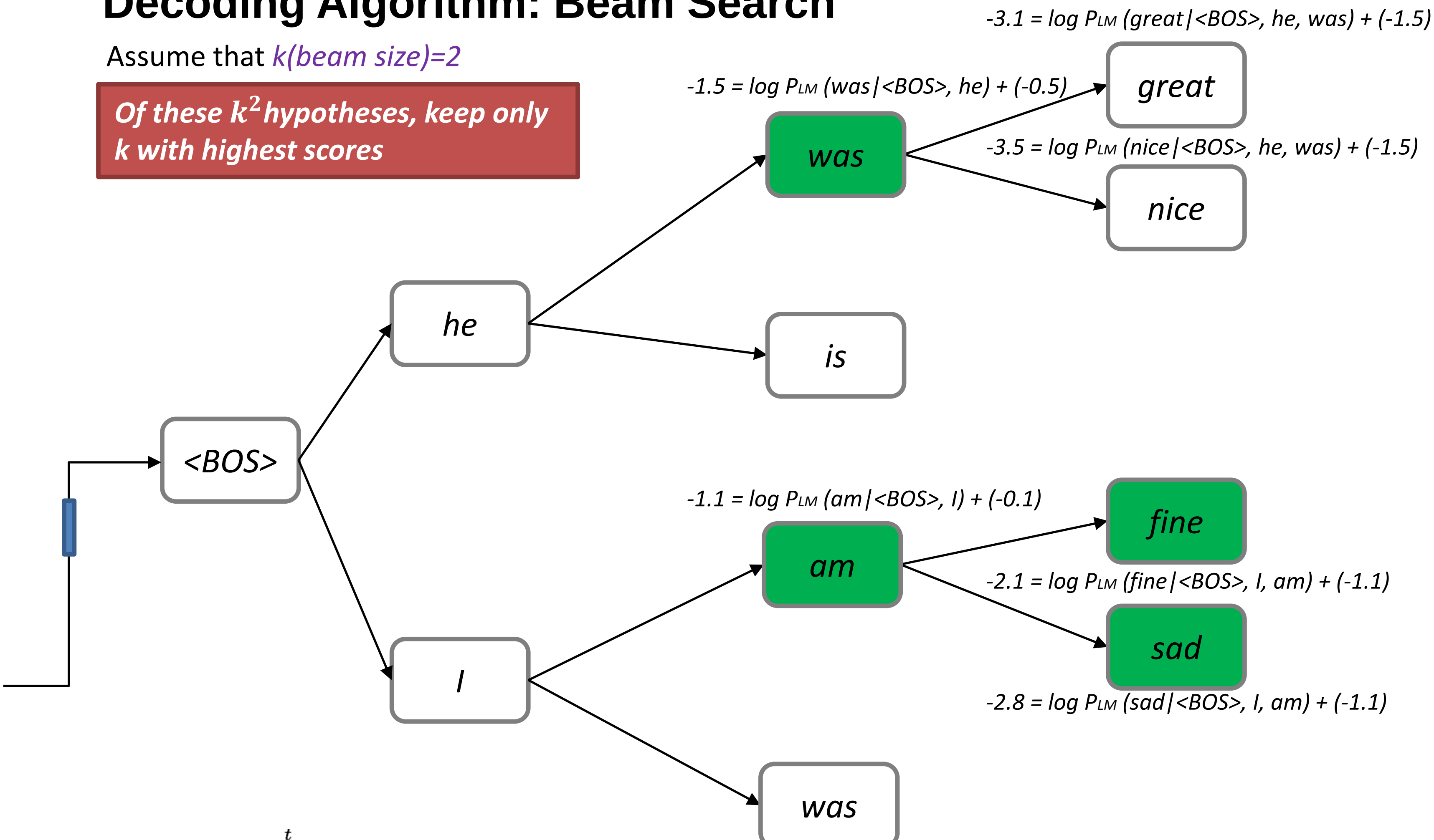


$$\text{score}(y_1,\ldots,y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i|y_1,\ldots,y_{i-1},x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Of these k^2 hypotheses, keep only k with highest scores

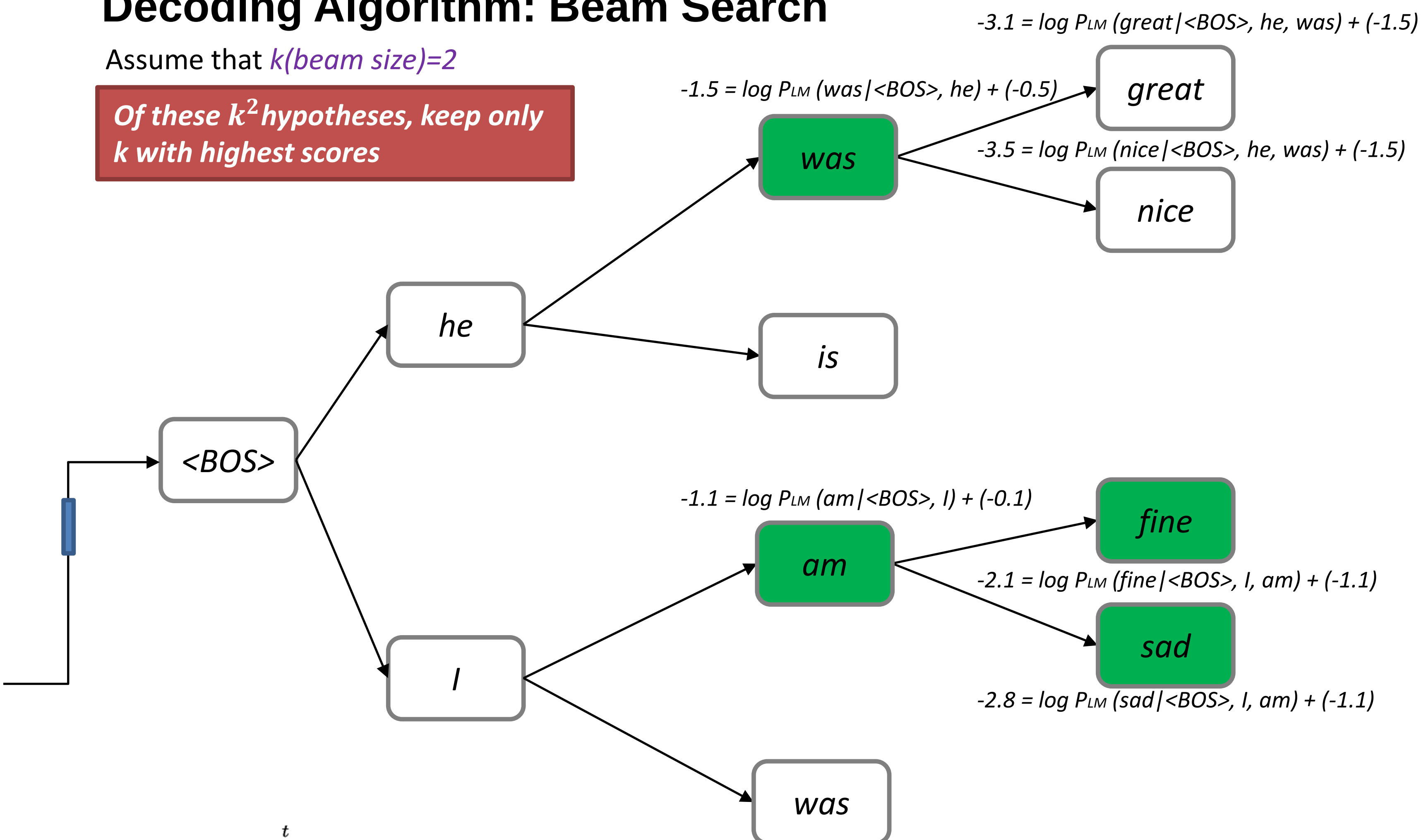


$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Of these k^2 hypotheses, keep only k with highest scores

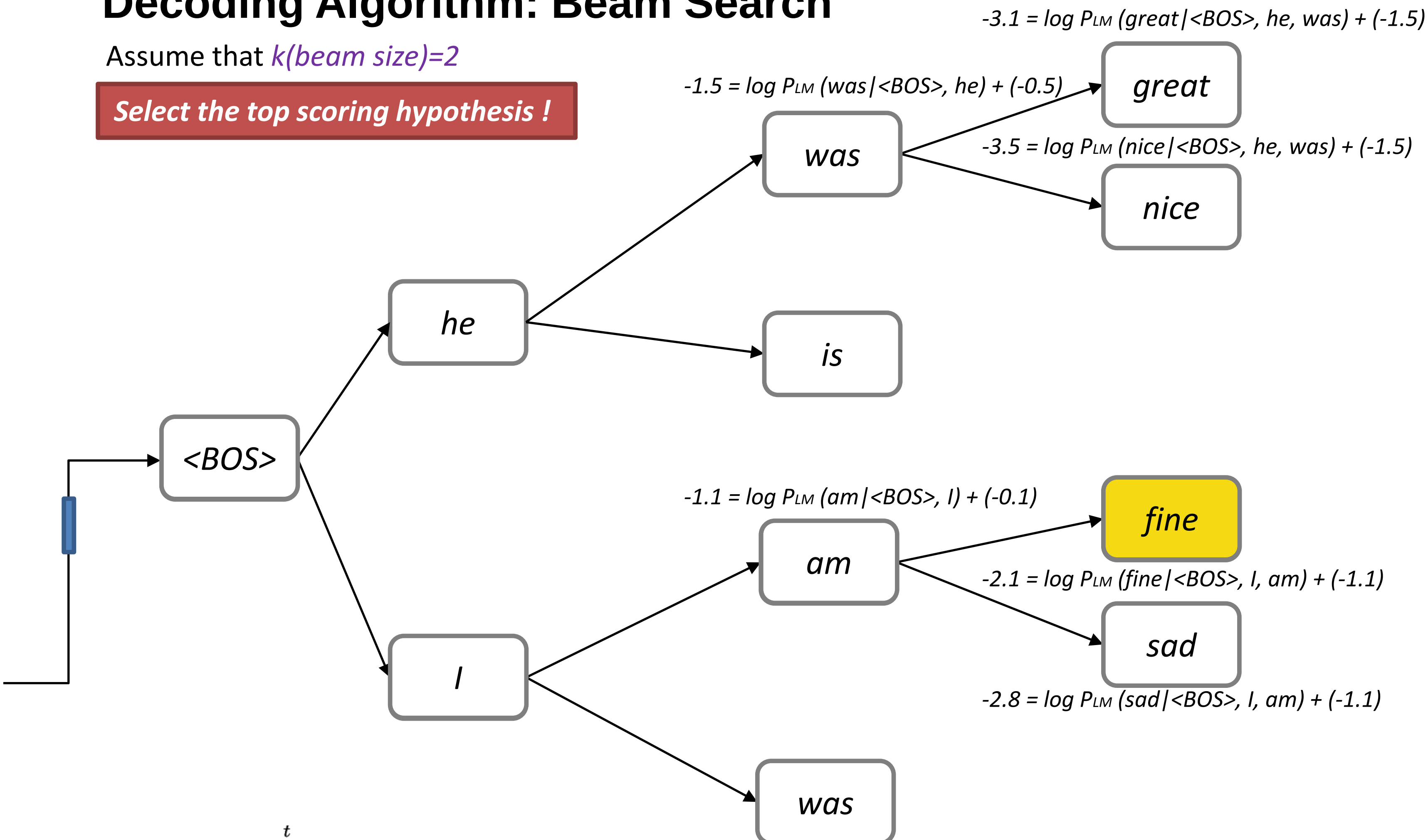


$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Decoding Algorithm: Beam Search

Assume that *k(beam size)=2*

Select the top scoring hypothesis !



$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

The effect of beam size k

- Small k has similar problems to greedy decoding (k=1)
  - Why?

- Large k means you consider more hypotheses
  - Solve the issues in greedy decoding
  - Produce, other issues:
    - Computationally expensive
    - In open-ended tasks like chit-chat dialogue, large k can make output more generic

4 Natural Language Generation

The effect of beam size k in chit chatbot

Human

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

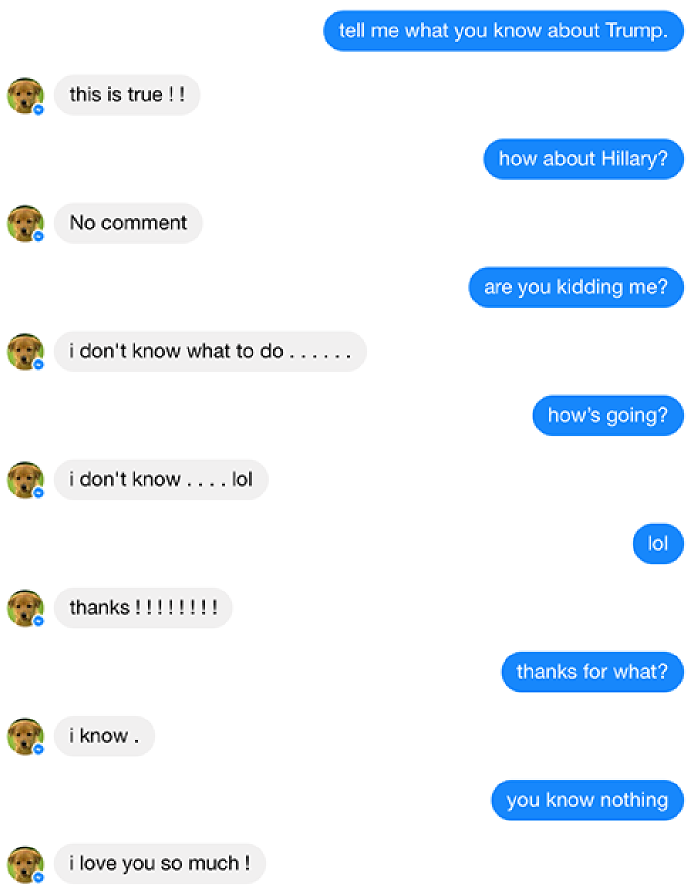
Machine Answer

Lower beam size
More on topic but non-sensical

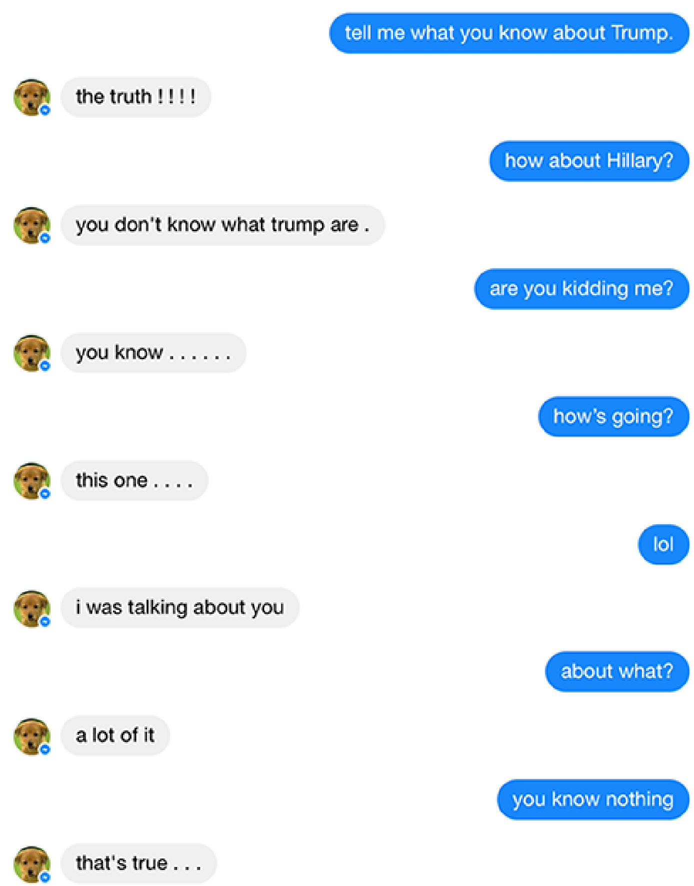
Higher beam size
Converges to safe, “correct” response, but it’s generic and less relevant

THE UNIVERSITY OF SYDNEY

The effect of beam size k in chit chatbot



Beam size=10



Beam size=10 and anti-language model

Sampling-based decoding

Pure sampling

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

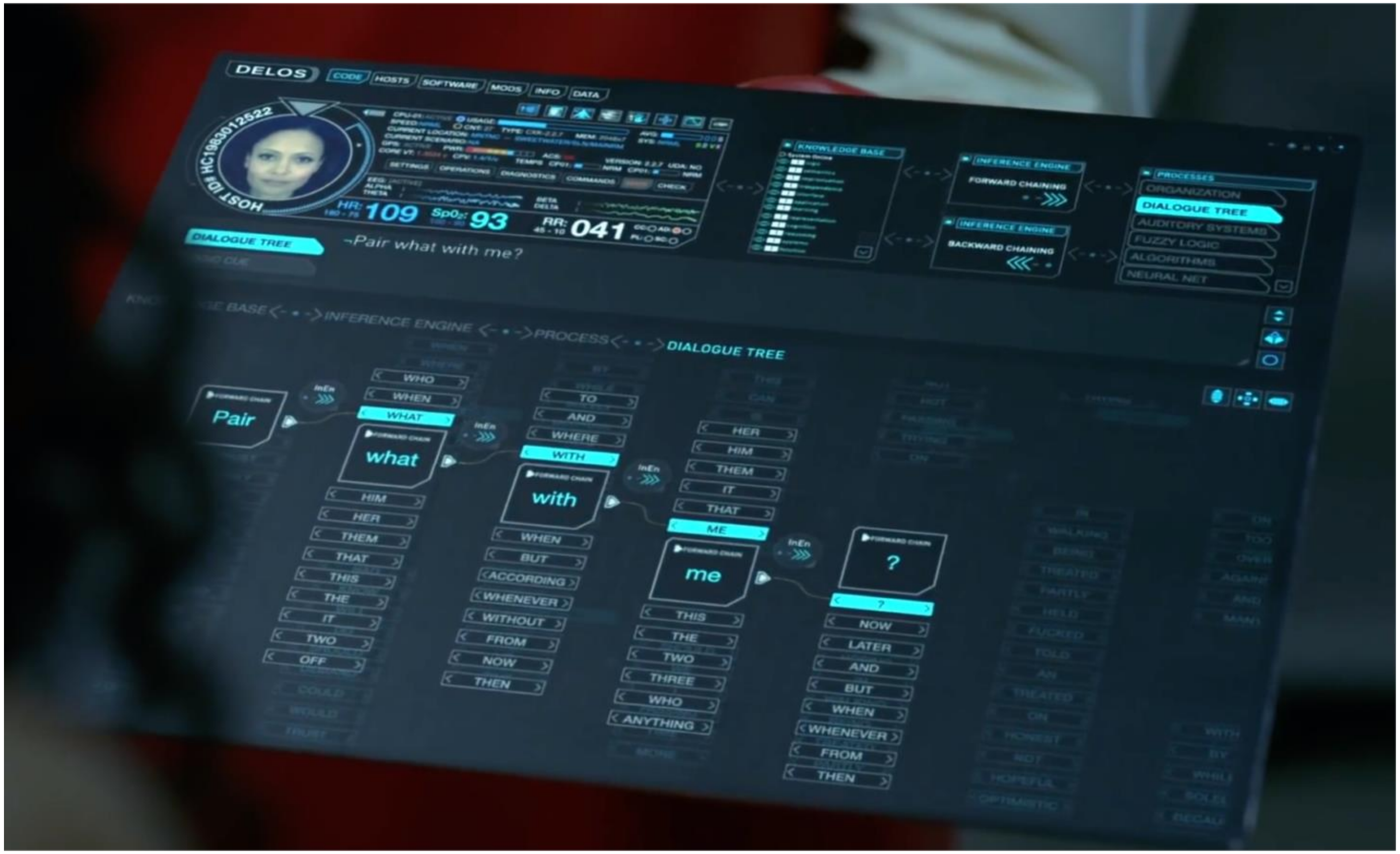
Top-n sampling*

- On each step t, randomly sample from P_t , restricted to just the top-n most probable words
- Like pure sampling, but truncate the probability distribution
- n=1 is greedy search, n=V is pure sampling
- Increase n to get more diverse/risky output
- Decrease n to get more generic/safe output

**Usually called top-k sampling, but here we're avoiding confusion with beam size k*

Natural Language Generation

Dialog Tree from Westworld



https://www.youtube.com/watch?v=ZnxJRYit44k&feature=youtu.be&t=28

0 LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

1. Language Model
2. Traditional Language Model
3. Neural Language Model
4. Natural Language Generation
5. **NLG Tasks**
6. Language Model and NLG Evaluation

NLG Tasks

Language Modeling in Natural Language Generation

Natural Language Generation Tasks

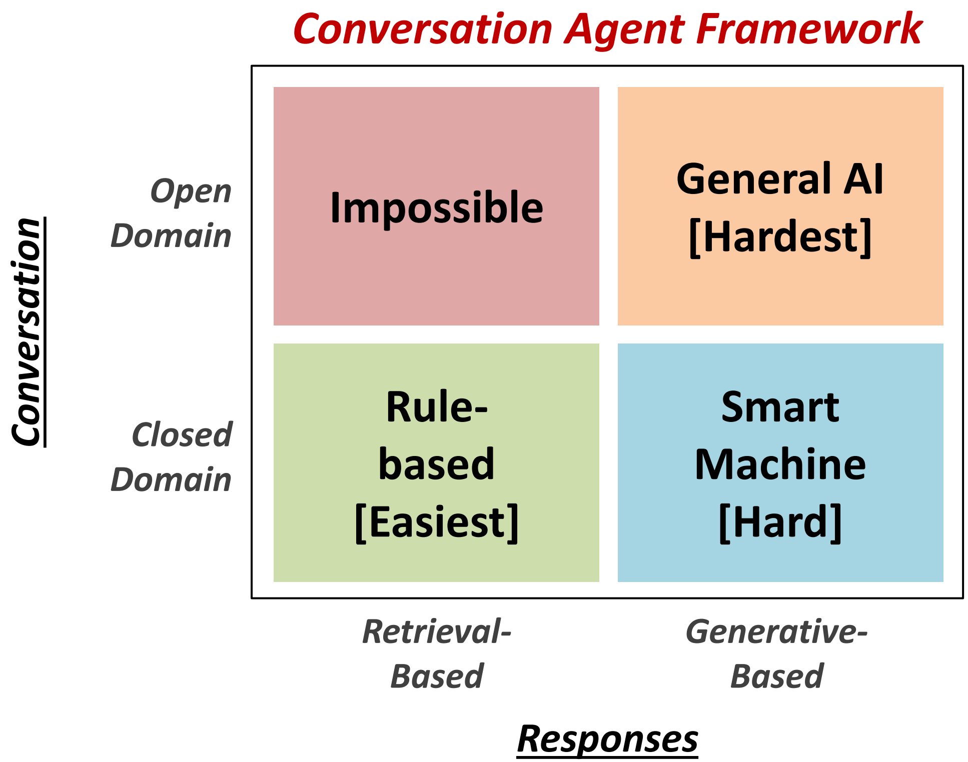
- Dialogue (chit chat and goal-oriented conversational agent)
  x=dialogue history, y=next utterance

- Abstractive Summarisation
  x=input text, y=summarized text

5 NLG Tasks

Dialog: Conversational Agent

A conversational agent is a software program which interprets and responds to statements made by users in ordinary natural language. It integrates computational linguistics techniques with communication over the internet



Conversational Agent

A conversational agent is a software program which interprets and responds to statements made by users in ordinary natural language. It integrates computational linguistics techniques with communication over the internet

Goal-oriented Conversational Agent

Designed for a particular task, utilizing short conversations to get information from the user to help complete this task

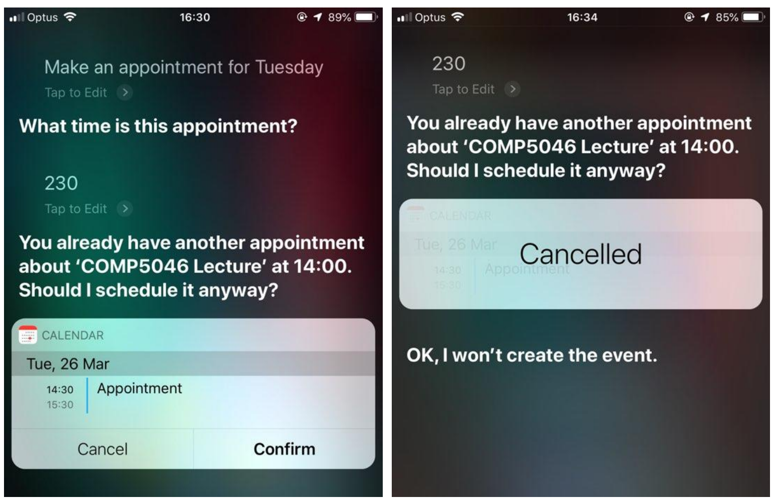
Chatbots (Chat-oriented Conversational Agent)

Designed to handle full conversations, mimicking the unstructured flow of a human to human conversation

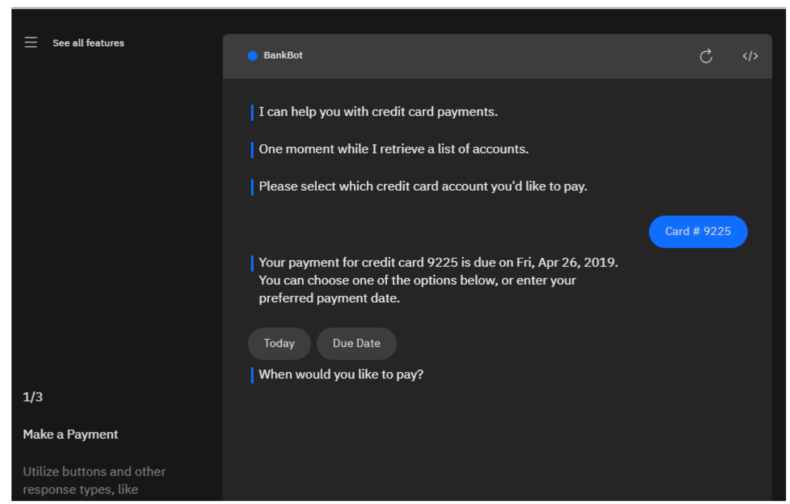
5 NLG Tasks

Goal-oriented Conversational Agent

Designed for a particular task, utilizing short conversations to get information from the user to help complete this task



Apple Siri



IBM Watson BankBot

Goal-oriented Conversational Agent

Frame-based Approach

- Based on a "**domain ontology**"
  A knowledge structure representing user intentions

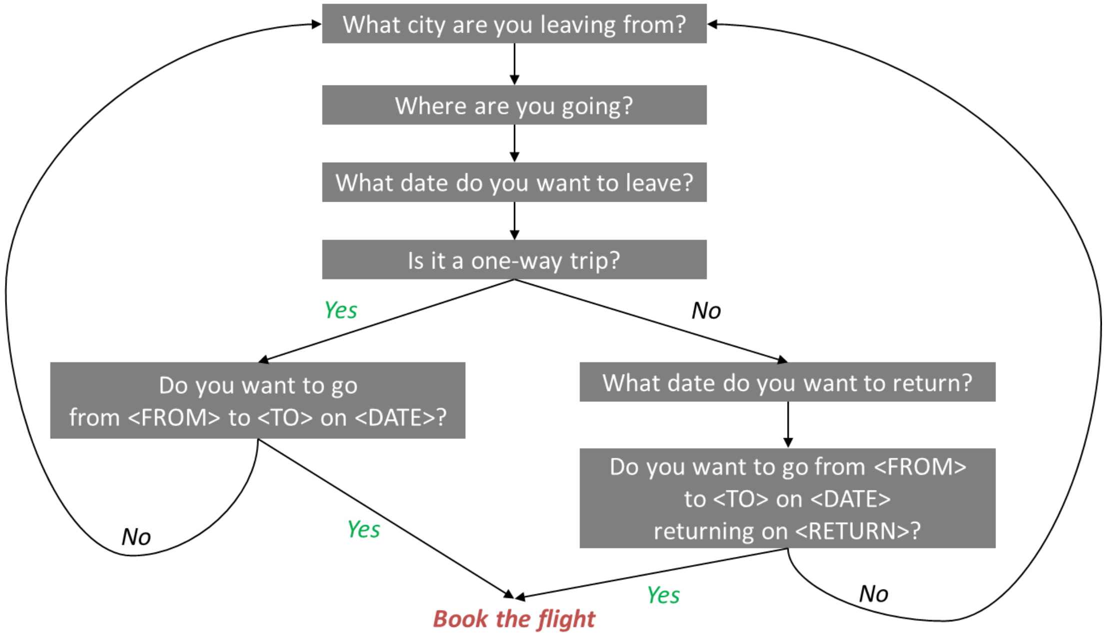
- One or more Frame
  Each a collection of **slots**
  Each slot having a **value**
  A set of **slots**, to be filled with information of a given **type**
  Each associated with a **question** to the user

Slot	***Type***	***Question***
ORIGIN	*city*	*What city are you leaving from?*
DEST	*city*	*Where are you going?*
DEPT DATE	*date*	*What day would you like to leave?*
DEPT TIME	*time*	*What time would you like to leave?*
AIRLINE	*line*	*What is your preferred airline?*

Goal-oriented Conversational Agent

Dialogue is structured in a sequence of predetermined utterance

- Ask the user for a departure city
- Ask for a destination city
- Ask for a time
- Ask whether the trip is round--trip or not



Goal-oriented Conversational Agent

- System completely controls the conversation with the user.
- It asks the user a series of questions
- Ignoring (or misinterpreting) anything the user says that is not a direct answer to the system's questions

Dialogue Initiative

Systems that control conversation like this are:
system initiative or ***single initiative***

Initiative: who has control of conversation

In normal human to human dialogue,
initiative shifts back and forth between participants

NLG Tasks

System Initiative

System completely controls the conversation

- *Simple to build*
- *User always knows what they can say next*
- *System always knows what user can say next*
- *Good for Very Simple tasks (entering a credit card, booking a flight)*

- *Too limited: does not generate any new text, they just pick a response from a fixed set*
- *A lot of hard coded rules have to be written so not much intelligent*

System Initiative: Issue

"Hi, I'd like to fly from Sydney Tuesday morning; I want a flight from Melbourne to Perth one way leaving after 5 p.m. on Wednesday."

- Answering more than one question in a sentence

Mixed Initiative

Conversational initiative can shift between system and user

"Hi, I'd like to fly from Sydney Tuesday morning; I want a flight from Melbourne to Perth one way leaving after 5 p.m. on Wednesday."

A kind of **mixed initiative**

- use the structure of the **frame** to guide dialogue

- System asks questions of user, filling any slots that user specifies
  - When frame is filled, do database query
- If user answers 3 questions at once, system can fill 3 slots and not ask these questions again!

Mixed Initiative

- There are many ways to represent the meaning of sentences
- For speech dialogue systems, most common approach is "Frame and slot semantics".

"Show me morning flights from Sydney to Perth on Tuesday."

DOMAIN:	*AIR-TRAVEL*
INTENT:	*SHOW-FLIGHTS*
ORIGIN-CITY:	*Sydney*
ORIGIN-DATE:	*Tuesday*
ORIGIN-TIME:	*morning*
DEST-CITY:	*Perth*

SIRI: Condition-Action Rules

Active Ontology: Relational network of concepts

- **Data structures**: a meeting has:
  - a date and time,
  - a location,
  - a topic
  - a list of attendees
- **Rule sets** that perform actions for concepts
  - The date concept turns string
  - Monday at 2pm into
  - Date object *date(DAY,MONTH,YEAR,HOURS,MINUTES)*

Rule: Condition + Action

Improvements to the Rule-based Approach

Machine Learning classifiers to map words to semantic frame-fillers

Given a set of labeled sentences
- "I want to fly to Sydney on Tuesday"
- Destination: Sydney
- Depart-date: Tuesday

Build a classifier to map from one to the other

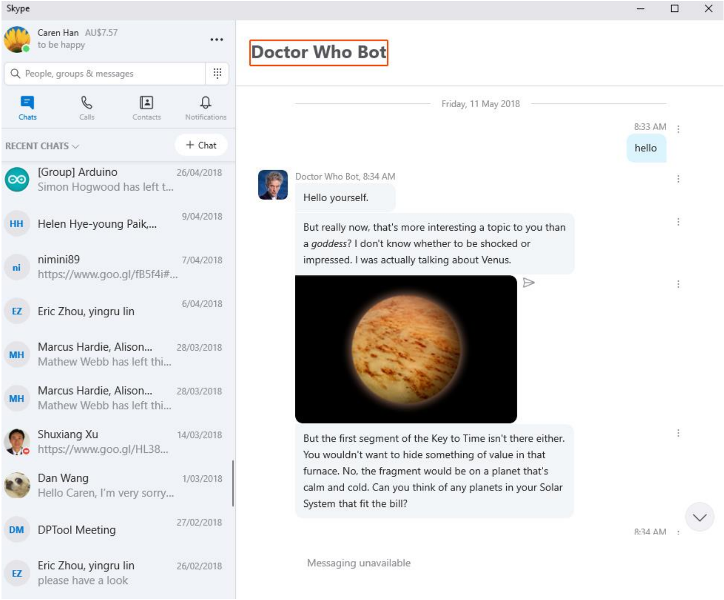
Requirements: Lots of Labeled Data

Conversational Agent

A conversational agent is a software program which interprets and responds to statements made by users in ordinary natural language. It integrates computational linguistics techniques with communication over the internet

Chatbot

Designed to handle full conversations, mimicking the unstructured flow of a human to human conversation



5 NLG Tasks

Chatbot

Designed to handle full conversations, mimicking the unstructured flow of a human to human conversation

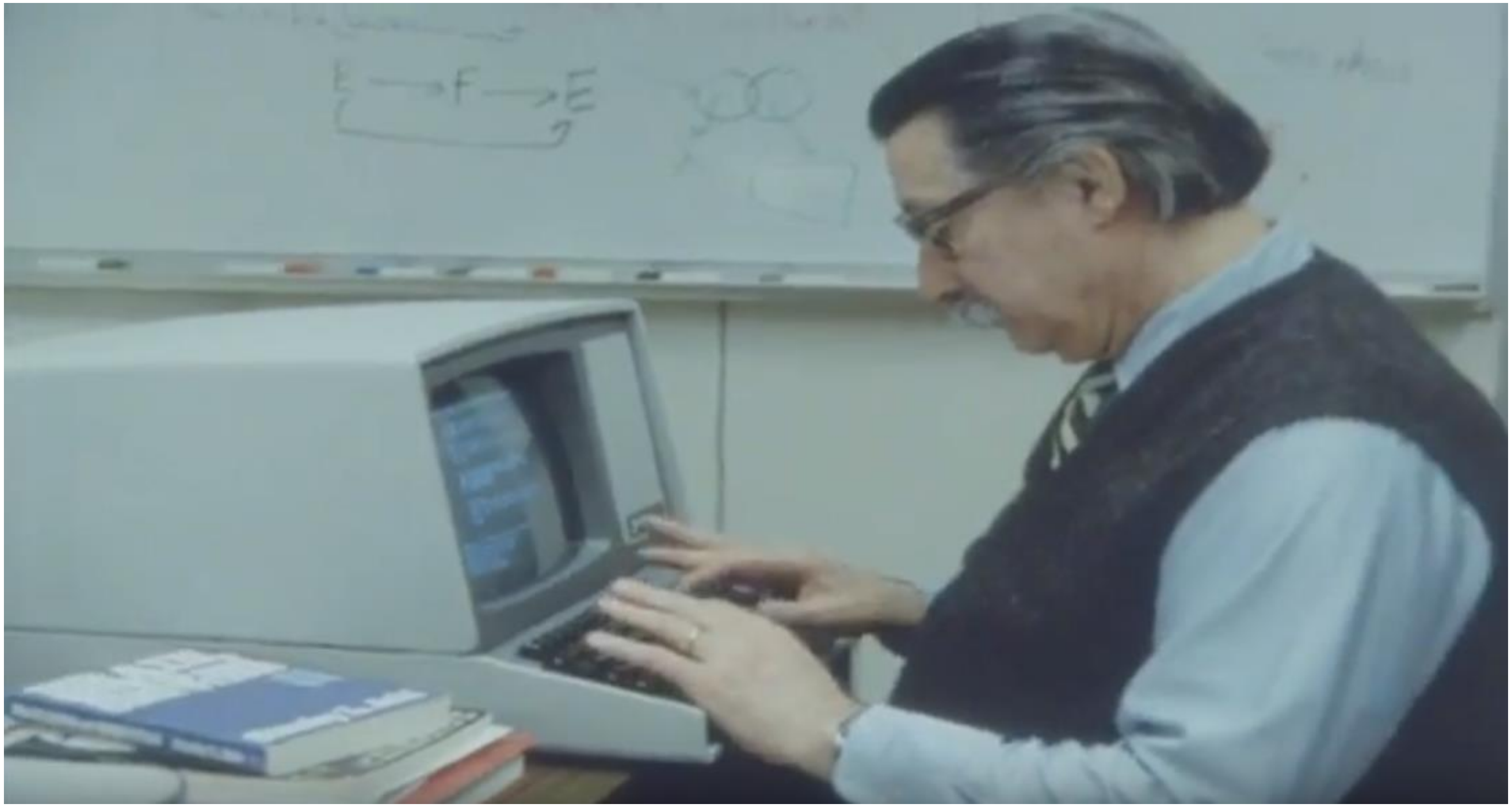
Rule-based

- Pattern-Action Rules (Eliza)
- Pattern-Action Rules + A mental model (Parry)

Corpus-based (from large chat corpus)

- Information Retrieval
- Deep Neural Networks

Chatbot: Eliza (1966)



Try Eliza

http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm

https://playclassic.games/game/play-eliza-online/play/

Chatbot: Eliza (1966)

Domain: Rogerian Psychology Interview

- Draw the patient out by reflecting patient's statements back at them
- Rare type of conversation in which one can "assume the pose of knowing almost nothing of the real world"

Patient: "I went for a long boat ride"
Psychiatrist: "Tell me about boats"

- You don't assume she didn't know what a boat is
- You assume she had some conversational goal

NLG Tasks

Chabot: Eliza (1966)

Pattern matching

if the input matches

(first bunch of words) "you" (second bunch of words) "me"

response with

"What makes you think I" (second bunch of words) "you?"

if the input matches

"You are" (bunch of words)

response with

"So, I'm" (bunch of words) ", am I?"

Very basic reconstruction rules

"me" → "you"

"my" → "your" *etc.*

Chatbot: Eliza (1966)

Some programmed responses to special keywords

if the word “mother” appears anywhere, reply with
“Don’t you talk about my mother”

Randomisation to avoid getting stuck in a rut

When all else fails, some stock responses,

“Tell me more”
“Fascinating”
“I see”

Chatbot: Parry (1972)

Same pattern--response structure as Eliza

Persona

- 28--year--old single man, post office clerk
- no siblings and lives alone
- Sensitive about his physical appearance, his family, his religion, his education and the topic of sex.
- Hobbies are movies and gambling on horseracing,
- Recently attacked a bookie, claiming the bookie did not pay off in a bet.
- Afterwards worried about possible underworld retaliation
- Eager to tell his story to non--threating listeners.

Chatbot: Parry (1972)

⟨OTHER'S INTENTION⟩ ← ⟨MALEVOLENCE⟩ | ⟨BENEVOLENCE⟩ | ⟨NEUTRAL⟩

MALEVOLENCE-DETECTION RULES

1. ⟨malevolence⟩ ← ⟨mental harm⟩ | ⟨physical threat⟩
2. ⟨mental harm⟩ ← ⟨humiliation⟩ | ⟨subjugation⟩
3. ⟨physical threat⟩ ← ⟨direct attack⟩ | ⟨induced attack⟩
4. ⟨humiliation⟩ ← ⟨explicit insult⟩ | ⟨implicit insult⟩
5. ⟨subjugation⟩ ← ⟨constraint⟩ | ⟨coercive treatment⟩
6. ⟨direct attack⟩ ← CONCEPTUALIZATIONS ([you get electric shock], [are you afraid mafia kill you?])
7. ⟨induced attack⟩ ← CONCEPTUALIZATIONS ([I tell mafia you], [does mafia know you are in hospital?])
8. ⟨explicit insult⟩ ← CONCEPTUALIZATIONS ([you are hostile], [you are mentally ill?])
9. ⟨implicit insult⟩ ← CONCEPTUALIZATIONS ([tell me your sexlife], [are you sure?])
10. ⟨constraint⟩ ← CONCEPTUALIZATIONS ([you stay in hospital], [you belong on locked ward])
11. ⟨coercive treatment⟩ ← CONCEPTUALIZATIONS ([I hypnotize you], [you need tranquilizers])

5 NLG Tasks

Chatbot

Rule-based

- Pattern-Action Rules (Eliza)
- Pattern-Action Rules + A mental model (Parry)

Corpus-based (from large chat corpus)

- **Information Retrieval**
- **Deep Neural Networks**

5 NLG Tasks

Information Retrieval (IR) based Chatbot

- Mine conversations of human chats or human-machine chats
  - Microblogs: Twitter etc.
  - Movie Dialogs
- With large corpus

Cleverbot

https://www.cleverbot.com/

Microsoft Xiaoice

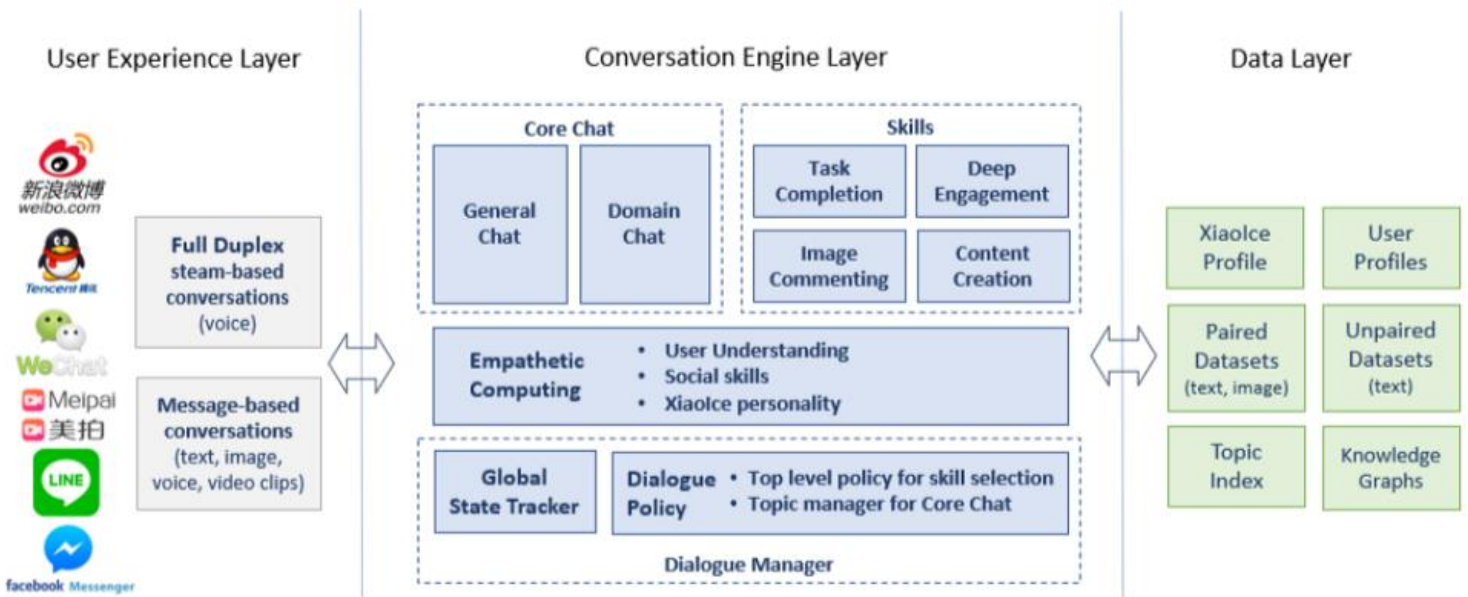
https://arxiv.org/pdf/1812.08989.pdf

Microsoft Tay

https://youtu.be/Lr4yi9onykg

Information Retrieval (IR) based Chatbot

Xiaoice



Information Retrieval (IR) based Chatbot

1. Return the response to the most similar turn
   - Take user's turn (q) and find a (tf-idf) similar turn t in the corpus C

     q="do you like Doctor Who"
     t="do you like Doctor Strange"

   - Grab whatever the response was to t.

$$r = response\left(\operatorname*{argmax}_{t \in C} \frac{q^T t}{||q||t||}\right)$$

Yes, love it!

2. Return the most similar turn

$$r = \operatorname*{argmax}_{t \in C} \frac{q^T t}{||q||t||}$$

Do you like Doctor Strangelove?

5 NLG Tasks

Information Retrieval (IR) based Chatbot

1. Also fine to use other features like user features, or prior turns

2. Or non-dialogue text
   - COBOT chatbot (Isbell et al., 2000)
   - sentences from the Unabomber Manifesto by Theodore Kaczynski, articles on alien abduction, the scripts of “The Big Lebowski” and “Planet of the Apes”.
3. Wikipedia text

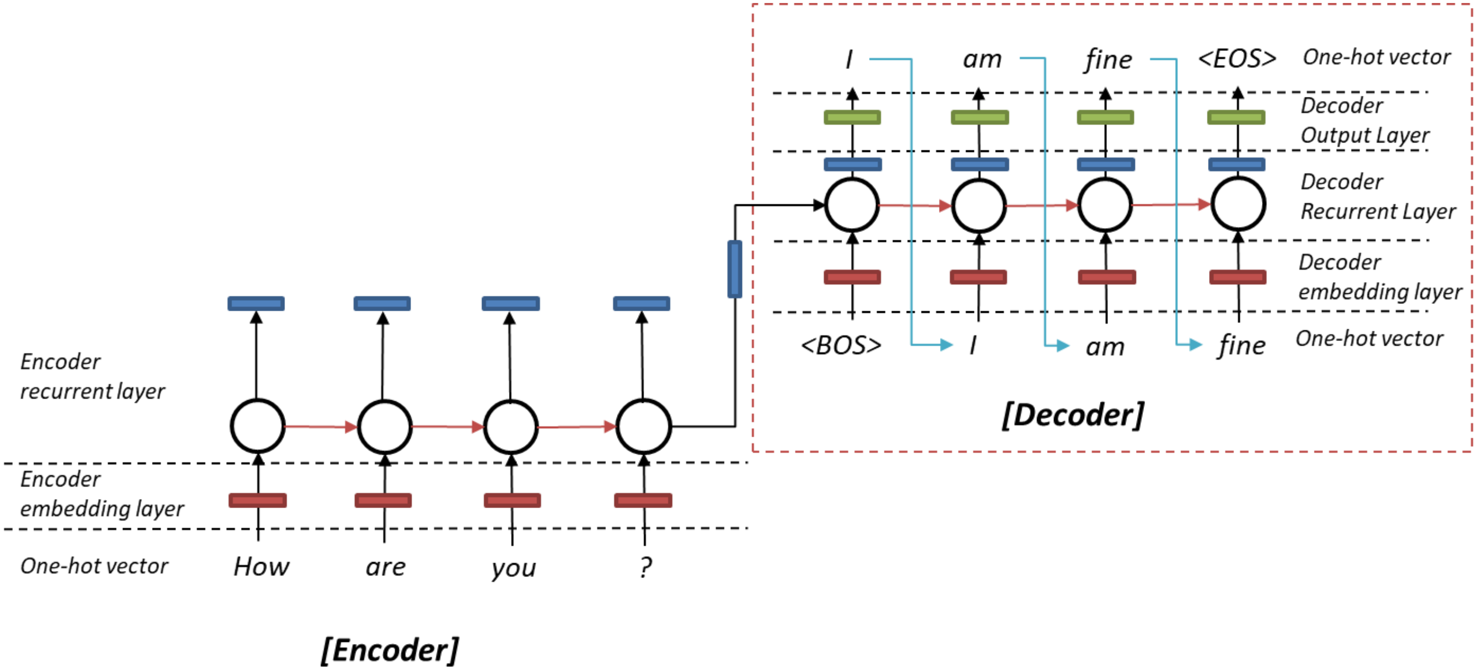
5 NLG Tasks

Deep-learning Chatbots

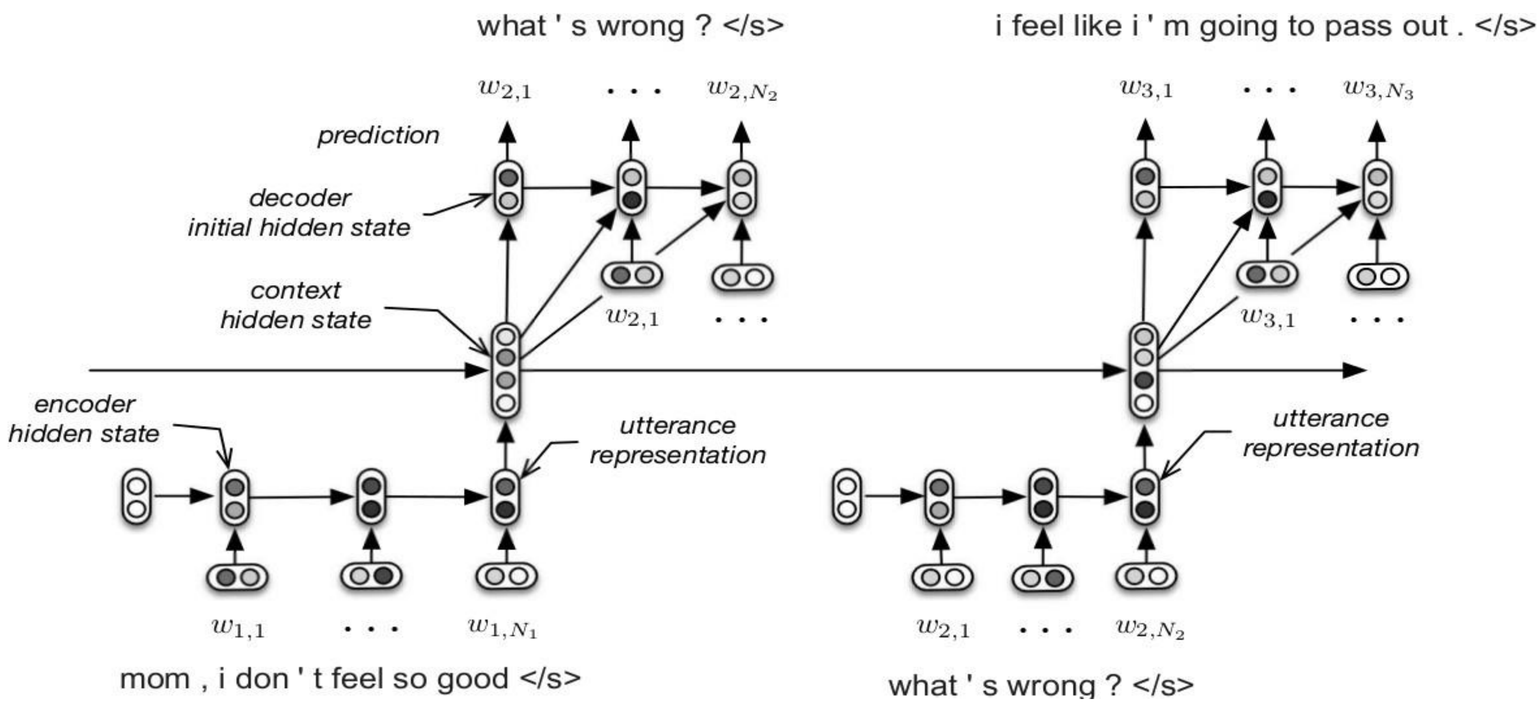
- Think of response generation as a task of transducing from the user's prior turn to the system's turn.
- Train on:
  - Movie Dialogs
  - Twitter Conversations

- Train a deep neural network
  - Map from user 1 turn to user 2 response

Seq2seq model architecture



Seq2seq model architecture



5 NLG Tasks

Deep learning chatbots

Trained on 127M Twitter context-message-response triples

#	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.

Neural based NLG in Dialog: Issue

- Problem: became apparent that a naïve application of standard seq2seq methods has serious pervasive deficiency for (chitchat) dialogue:
  - Either because it’s generic (e.g. “I don’t know”)
  - Or because changing the subject to something unrelated
  - Boring response
  - Repetition problem
  - Lack of consistent persona problem

What else do we have?

Template-based generation

- The most common approach in spoken natural language generation.
- In simplest form, words fill in slots:

"Flights from *ORIGIN* to *DEST* on *DEPT_DATE DEPT_TIME*. Just one moment please"

Slot	*Type*	*Question*
ORIGIN	*city*	*What city are you leaving from?*
DEST	*city*	*Where are you going?*
DEPT DATE	*date*	*What day would you like to leave?*
DEPT TIME	*time*	*What time would you like to leave?*
AIRLINE	*line*	*What is your preferred airline?*

- Most common NLG used in commercial systems
- Used in conjunction with concatenative TTS (text-to-speech) to make natural sounding output

Template-based generation

Pros
- Conceptually Simple: No specialized knowledge required to develop
- Tailored to the domain, so often good quality

Cons
- Lacks generality: Repeatedly encode linguistic rules (e.g. subject-verb agreement)
- Little variation in style
- Difficult to grow/maintain: Each utterance must be manually added

Improvement?
- Need deeper utterance representations
- Linguistic rules to manipulate them

5 NLG Tasks

Rule-based Generation

Content Planning

- What information must be communicated?
    - Content selection and ordering

Sentence Planning

- What words and syntactic constructions will be used for describing the content?
    - Aggretation: What elements can be grouped together for more natural-sounding, succinct output?
    - Lexicalisation: What word are used to express the various entities?

Realisation

- How is it all combined into a sentence that is syntactically and morphologically correct?

5 NLG Tasks

Rule-based Generation

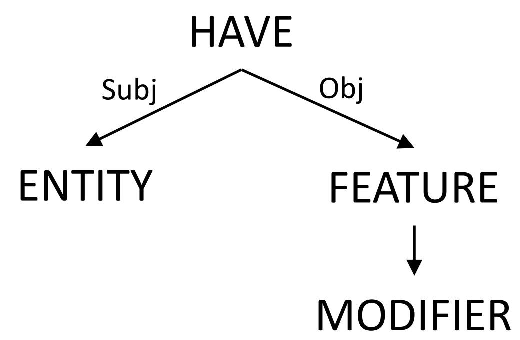
Assume that the dialog system need to tell the user about the restaurant

Content Planning

- Select Information ordering
  - has(sushitrain, crusine(bad))
  - has(sushitrain, decor(good))

Sentence Planning

- Choose ***syntactic templates***
- Choose lexicon
  - Bad → awful; crusine → food quality
  - Good → excellent; decor → décor
- Generate expressions
  - Entity → this restaurant



Realisation

- Choose correct verb: HAVE → has
- No article needed for feature names

"This restaurant has awful food quality but excellent décor"

5 NLG Tasks

Summary

Goal-oriented Conversational Agent:
- Ontology + hand-written rules for slot fillers
- Machine learning classifiers to fill slots

Chatbots:
- Simple rule-based systems
- IR-based: mine datasets of conversations.
- Neural net models with more data

The future…
- Need to acquire that data
- Integrate goal-based and chatbot-based systems

Summarisation: two strategies

Extractive Summarisation

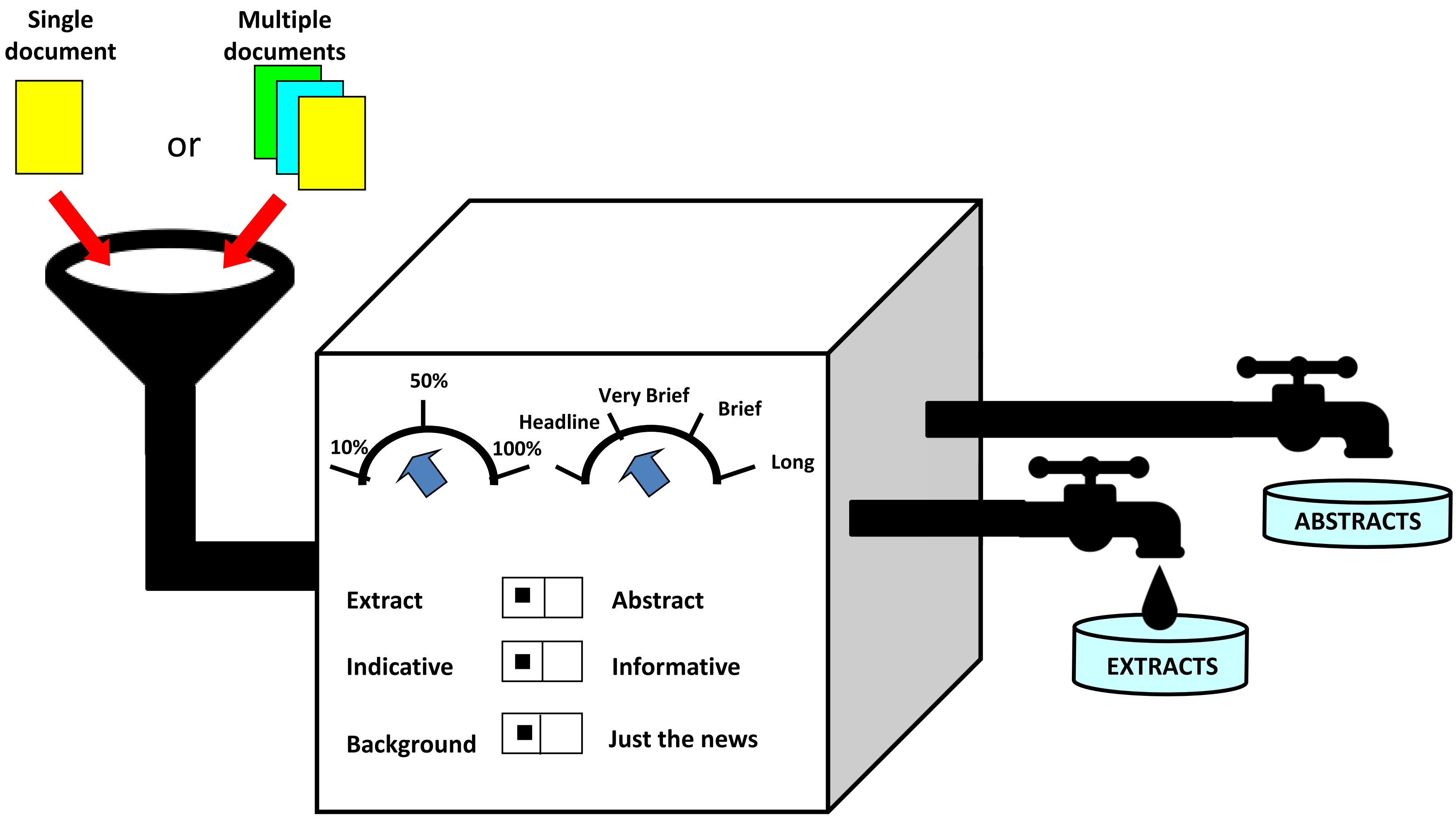
- Select parts (typically sentences) of the original text to form a summary.

Abstractive Summarisation

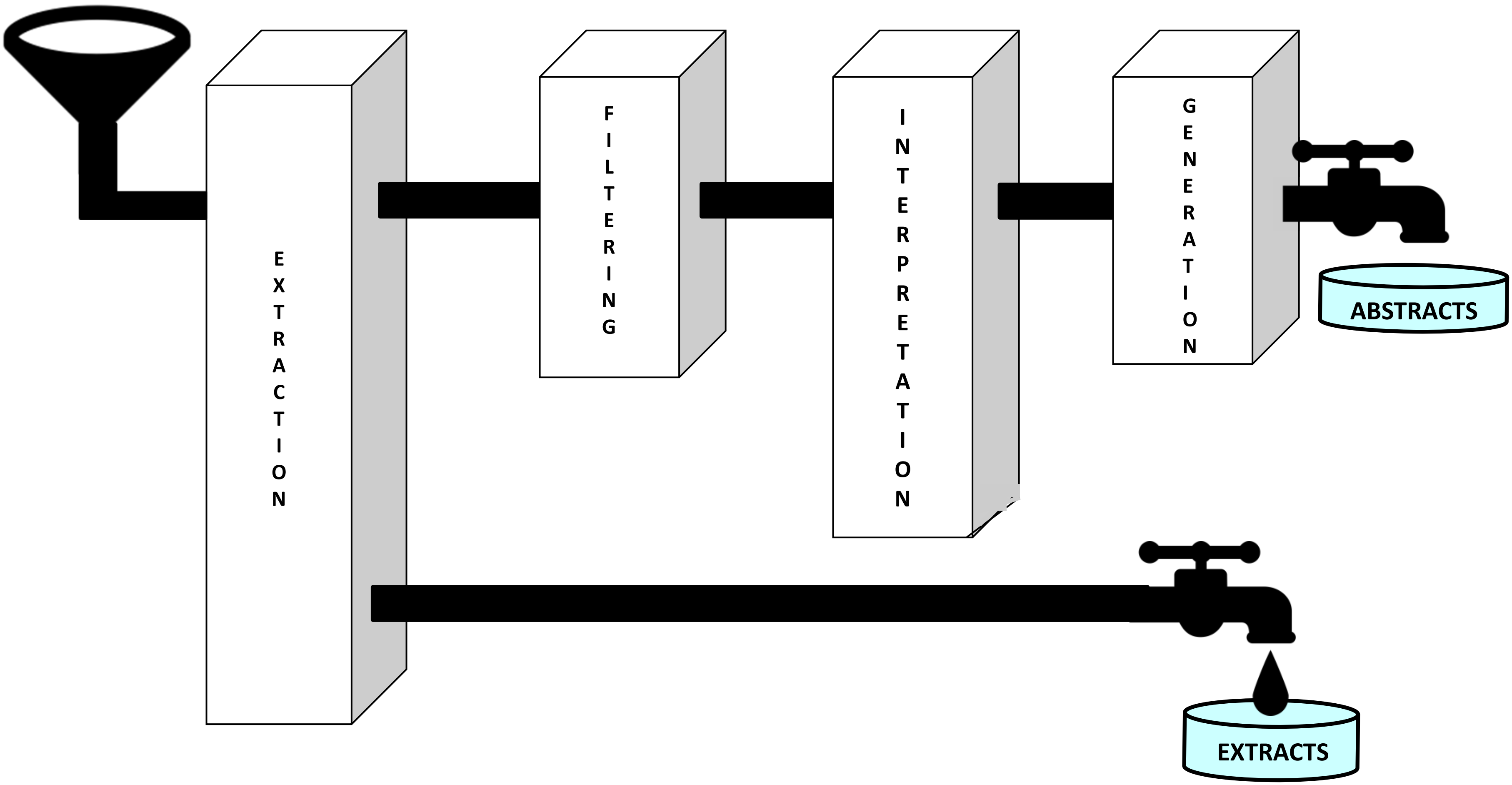
- *Generate new text using natural language generation techniques.*

NLG Tasks

Summarisation: two strategies



Summarisation: two strategies



0 LECTURE PLAN

Lecture 8: Language Model and Natural Language Generation

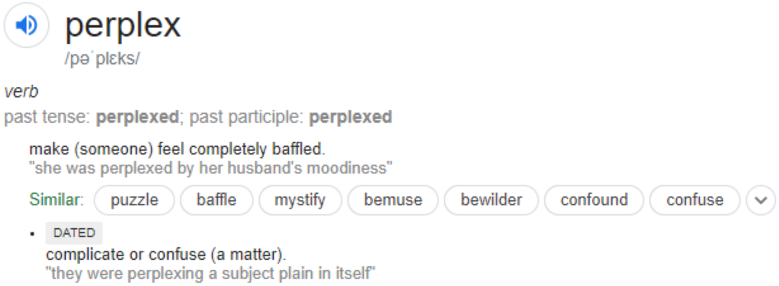
1. Language Model
2. Traditional Language Model
3. Neural Language Model
4. Natural Language Generation
5. NLG Tasks
6. **Language Model and NLG Evaluation**

6 Language Model and NLG Evaluation

How to evaluate the Language Model?

The standard evaluation metric for Language Models is ***perplexity****.*

Perplexed



perplex
/pəˈplɛks/

verb
past tense: **perplexed**; past participle: **perplexed**

make (someone) feel completely baffled.
"she was perplexed by her husband's moodiness"

Similar: puzzle baffle mystify bemuse bewilder confound confuse

- DATED
complicate or confuse (a matter).
"they were perplexing a subject plain in itself"

So, Lower Perplexity is better!

6 Language Model and NLG Evaluation

How to evaluate the Language Model?

The standard evaluation metric for Language Models is ***perplexity****.*

Perplexed

perplex

/pəˈplɛks/

verb

past tense: **perplexed**; past participle: **perplexed**

make (someone) feel completely baffled.
"she was perplexed by her husband's moodiness"

Similar: puzzle baffle mystify bemuse bewilder confound confuse

- DATED
  complicate or confuse (a matter).
  "they were perplexing a subject plain in itself"

$$\text{perplexity} = \prod_{t=1}^{T} \left(\frac{1}{P_{\text{LM}}(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})} \right)^{1/T}$$

Normalized by number of words

Inverse probability of corpus, according to Language Model

6 Language Model and NLG Evaluation

How to evaluate the Language Model?

Language Model Approaches Performance Evaluated by Facebook Research

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1	
RNN-2	
Sparse Non-negative Matrix Factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Joz	
2-layer LSTM-8	
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8

N-gram model

ural model

Can we use the perplexity for NLG Evaluation?

No. Captures how powerful your LM is,
but doesn't tell you anything about generation

Table 2. Comparison on 1B word in perplexity (lower the better). Note that Jozefowicz et al., uses 32 GPUs for training. We only use 1 GPU.

https://research.fb.com/building-an-efficient-neural-language-model-over-a-billion-words/

Language Model and NLG Evaluation

How to evaluate the Natural Language Generation?

Unfortunately, No automatic metrics to adequately capture overall quality

There are some metrics to capture particular aspects of generated text:

- *Fluency (compute probability - well-trained Language model)*
- *Correct style (Language Model 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*
- *Though these don’t measure overall quality, they can help us track some important qualities that we care about.*

Language Model and NLG Evaluation

How to evaluate the Natural Language Generation?

Human Evaluation

- *Human judgments are regarded as the gold standard*
- *Of course, we know that human eval is slow and expensive*
- *Supposing you do have access to human evaluation: Does human evaluation solve all of your problems?*

Humans …

- *are inconsistent*
- *can be illogical*
- *lose concentration*
- *misinterpret your question*
- *can't always explain why they feel the way they do*

Natural Language Generation: Long way to go

/ Reference

Reference for this lecture

- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. " O'Reilly Media, Inc.".
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Manning, C 2018, Natural Language Processing with Deep Learning, lecture notes, Stanford University

- Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055.
- Jiang, S., & de Rijke, M. (2018). Why are Sequence-to-Sequence Models So Dull? Understanding the Low-Diversity Problem of Chatbots. arXiv preprint arXiv:1809.01941.
- Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.