CS11-747 Neural Networks for NLP

Conditioned Generation

Graham Neubig

Carnegie Mellon University
Language Technologies Institute

Site
https://phontron.com/class/nn4nlp2021/

Language Models

- Language models are generative models of text

$$s \sim P(x)$$

"The Malfoys!" said Hermione.

Harry was watching him. He looked like Madame Maxime. When she strode up the wrong staircase to visit himself.

"I'm afraid I've definitely been suspended from power, no chance—indeed?" said Snape. He put his head back behind them and read groups as they crossed a corner and fluttered down onto their ink lamp, and picked up his spoon. The doorbell rang. It was a lot cleaner down in London.

Text Credit: Max Deutsch (https://medium.com/deep-writing/)

Conditioned Language Models

- Not just generate text, generate text according to some specification

Input *X*	Output *Y* (**Text**)	Task
Structured Data	NL Description	NL Generation
English	Japanese	Translation
Document	Short Description	Summarization
Utterance	Response	Response Generation
Image	Text	Image Captioning
Speech	Transcript	Speech Recognition

Formulation and Modeling

Calculating the Probability of a Sentence

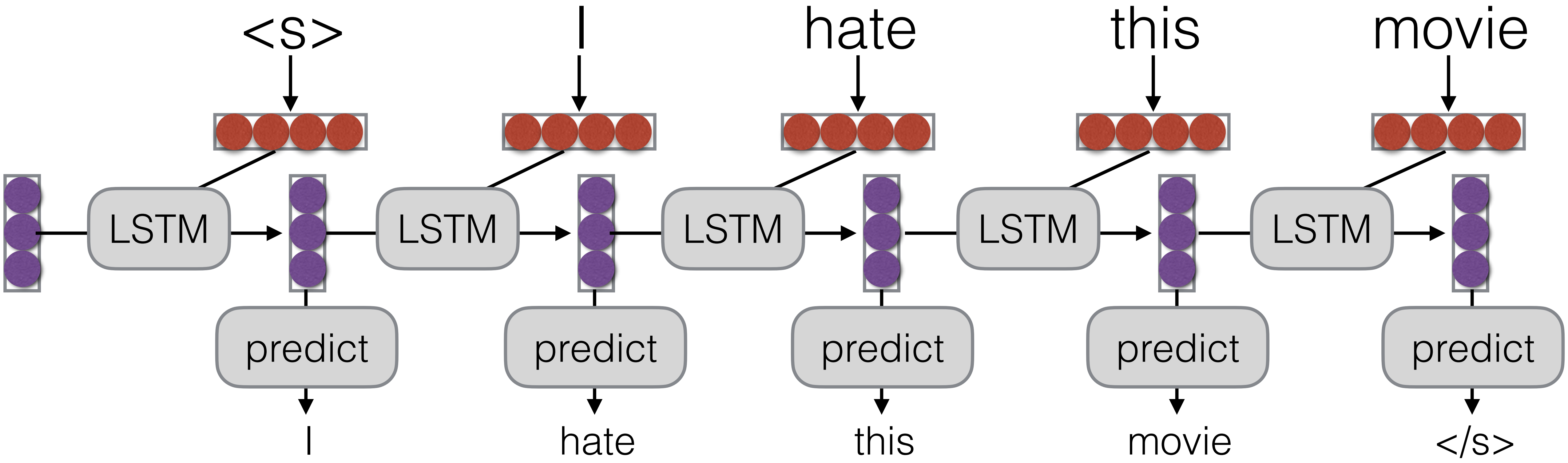
$$P(X) = \prod_{i=1}^{I} P(\underbrace{x_i}_{\text{Next Word}} \mid \underbrace{x_1, \ldots, x_{i-1}}_{\text{Context}})$$

Conditional Language Models

$$P(Y|X) = \prod_{j=1}^{J} P(y_j \mid \underline{X}, y_1, \ldots, y_{j-1})$$

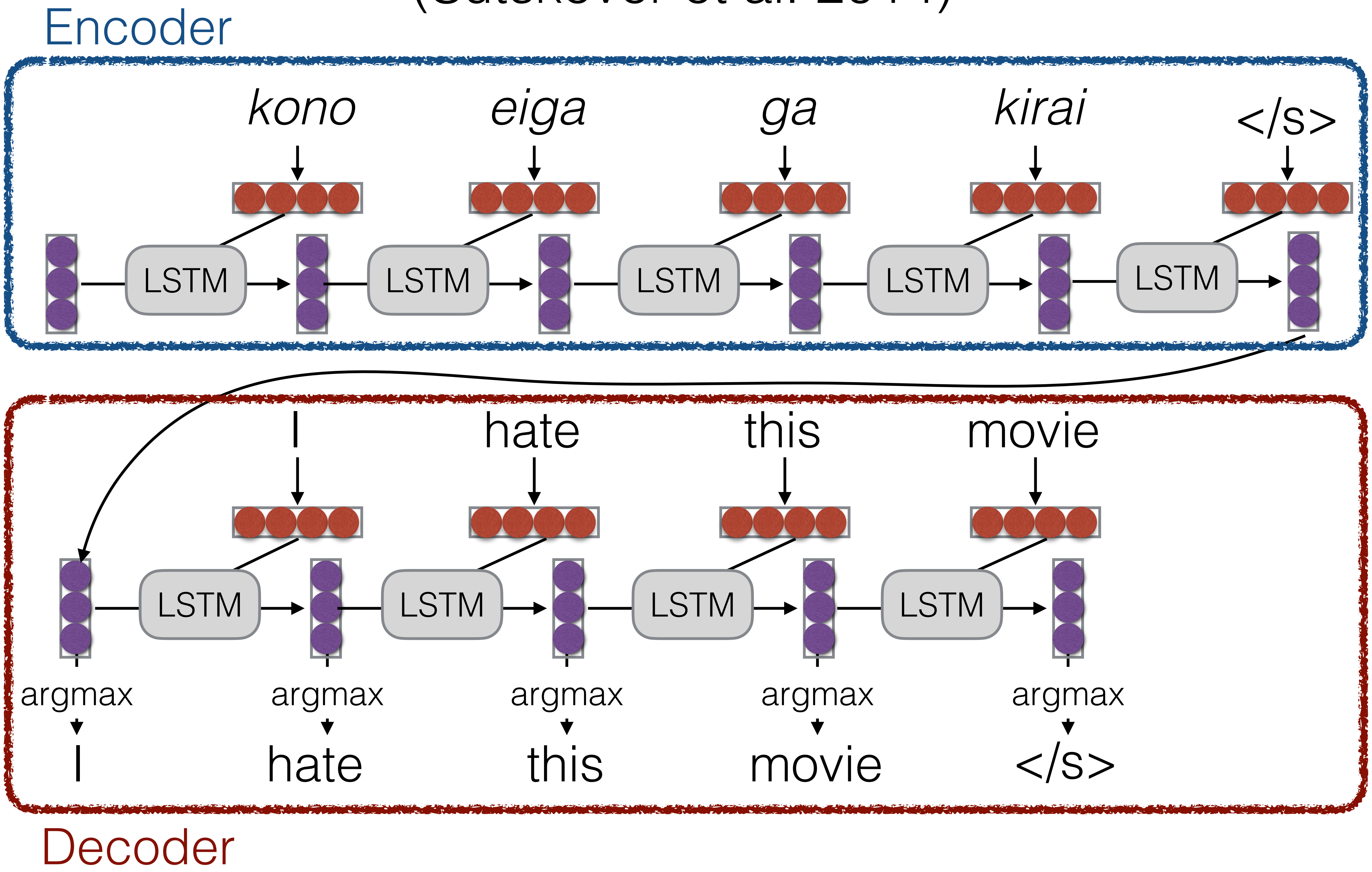
Added Context!

(One Type of) Language Model (Mikolov et al. 2011)



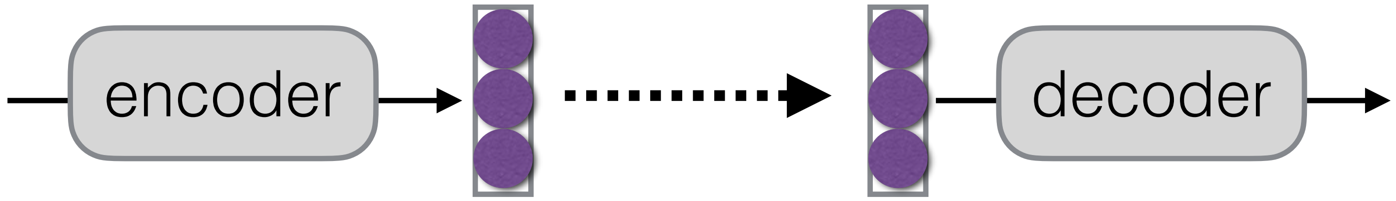
(One Type of) Conditional Language Model

(Sutskever et al. 2014)

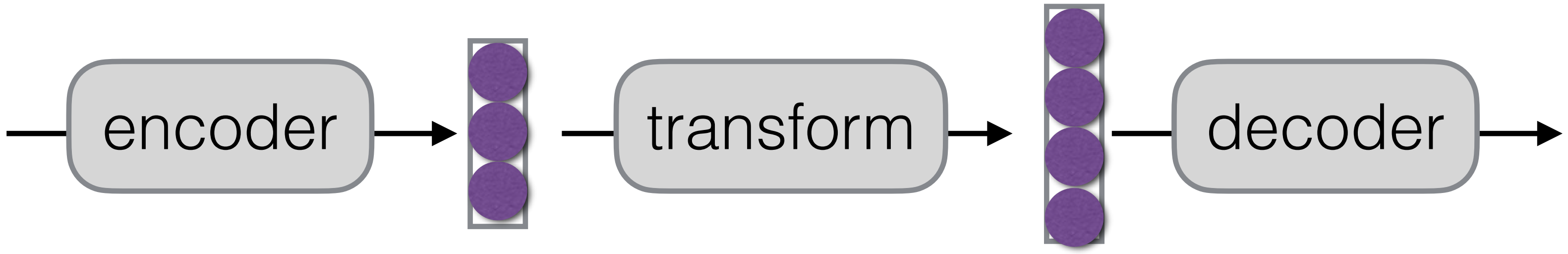


How to Pass Hidden State?

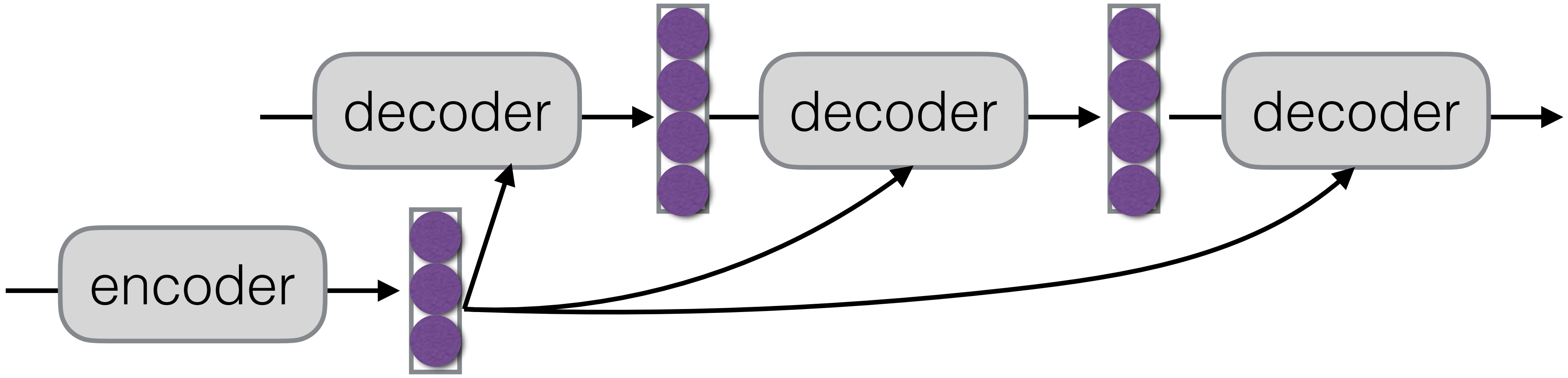
- Initialize decoder w/ encoder (Sutskever et al. 2014)



- Transform (can be different dimensions)



- Input at every time step (Kalchbrenner & Blunsom 2013)



Methods of Generation

The Generation Problem

- We have a model of P(Y|X), how do we use it to generate a sentence?
- Two methods:
  - **Sampling:** Try to generate a *random* sentence according to the probability distribution.
  - **Argmax:** Try to generate the sentence with the *highest* probability.

Ancestral Sampling

- **Randomly generate** words one-by-one.

```
while $y_{j-1}$ != “</s>”:
    $y_j \sim P(y_j \mid X, y_1, \ldots, y_{j-1})$
```

- An **exact method** for sampling from P(X), no further work needed.

Greedy Search

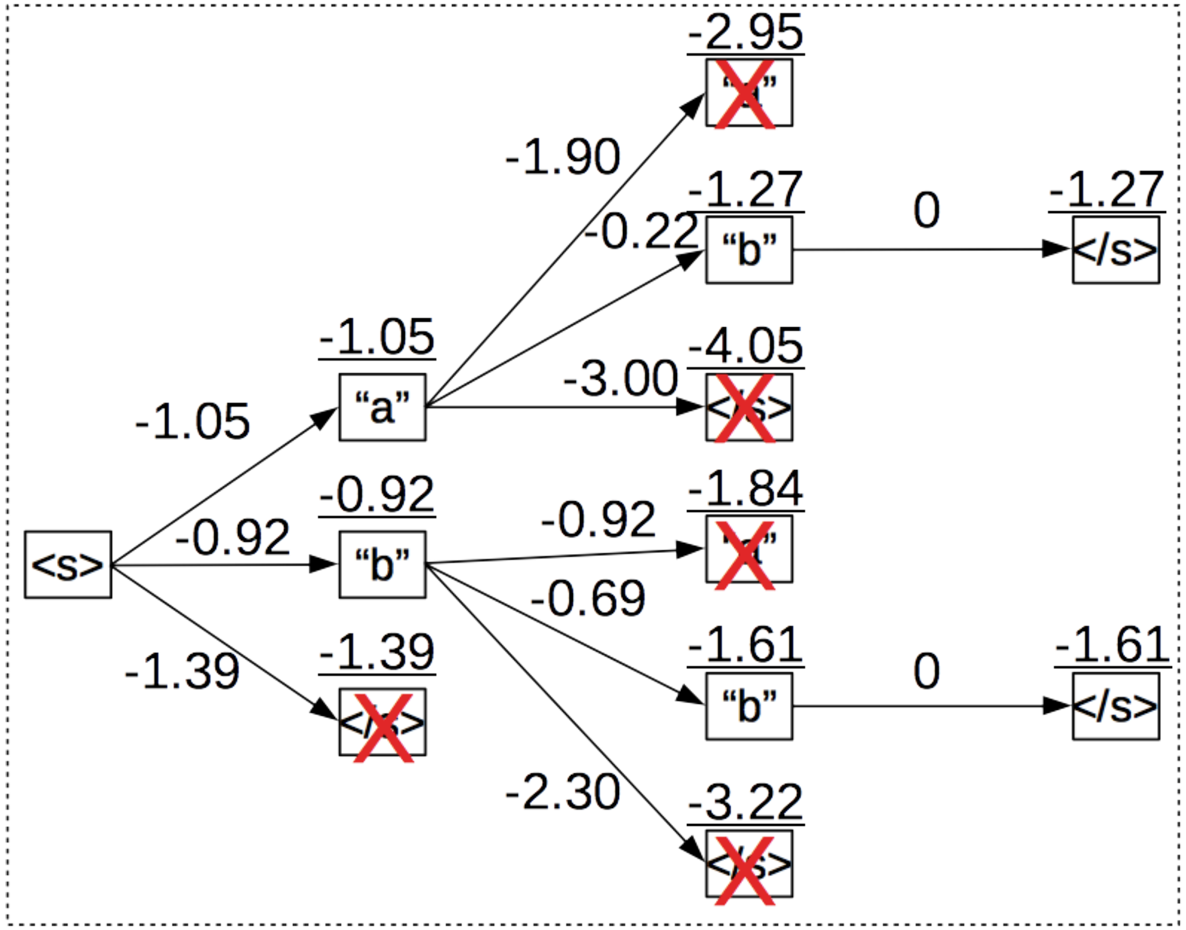
- One by one, pick the single highest-probability word

```
while $y_{j-1}$ != “</s>”:
  $y_j = \text{argmax}\ P(y_j \mid X, y_1, \ldots, y_{j-1})$
```

- **Not exact, real problems:**
  - Will often generate the “easy” words first
  - Will prefer multiple common words to one rare word

Beam Search

- Instead of picking one high-probability word, maintain several paths



- Some in reading materials, more in a later class

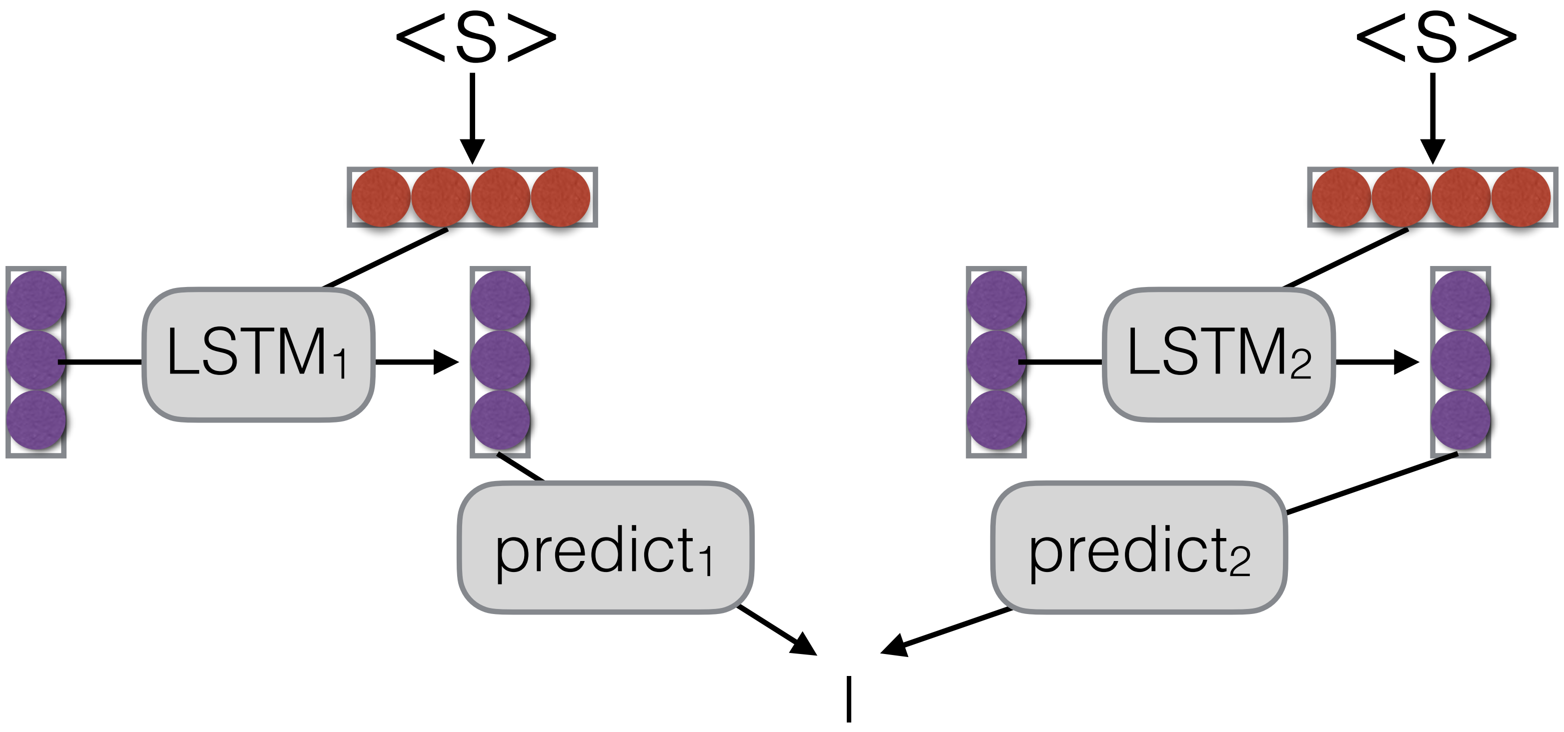
Let’s Try it Out!

`enc_dec.py`

Model Ensembling

Ensembling

- Combine predictions from multiple models



- Why?
  - Multiple models make somewhat uncorrelated errors
  - Models tend to be more uncertain when they are about to make errors
  - Smooths over idiosyncrasies of the model

Linear Interpolation

- Take a weighted average of the M model probabilities

$$P(y_j \mid X, y_1, \ldots, y_{j-1}) =$$

$$\sum_{m=1}^{M} \underbrace{P_m(y_j \mid X, y_1, \ldots, y_{j-1})}_{\text{Probability according to model } m} \underbrace{P(m \mid X, y_1, \ldots, y_{j-1})}_{\text{Probability of model } m}$$

- Second term often set to uniform distribution 1/M

Log-linear Interpolation

- Weighted combination of log probabilities, normalize

$$P(y_j \mid X, y_1, \ldots, y_{j-1}) = \underbrace{\text{softmax}}_{\text{Normalize}} \left(\sum_{m=1}^{M} \underbrace{\lambda_m(X, y_1, \ldots, y_{j-1})}_{\text{Interpolation coefficient for model } m} \underbrace{\log P_m(y_j \mid X, y_1, \ldots, y_{j-1})}_{\text{Log probability of model } m} \right)$$

- Interpolation coefficient often set to uniform distribution 1/M

Linear or Log Linear?

- Think of it in logic!
- **Linear:** “Logical OR”
  - the interpolated model likes any choice that a model gives a high probability
  - use models with models that capture different traits
  - necessary when any model can assign zero probability
- **Log Linear:** “Logical AND”
  - interpolated model only likes choices where all models agree
  - use when you want to restrict possible answers

Parameter Averaging

- **Problem:** Ensembling means we have to use *M* models at test time, increasing our time/memory complexity
- Parameter averaging is a cheap way to get some good effects of ensembling
- Basically, write out models several times near the end of training, and take the average of parameters

Ensemble Distillation (e.g. Kim et al. 2016)

- **Problem:** parameter averaging only works for models within the same run
- Knowledge distillation trains a model to **copy the ensemble**
    - Specifically, it tries to match the description over predicted words
    - Why? We want the model to make the same mistakes as an ensemble
- Shown to increase accuracy notably

Stacking

- What if we have two very different models where prediction of outputs is done in very different ways?

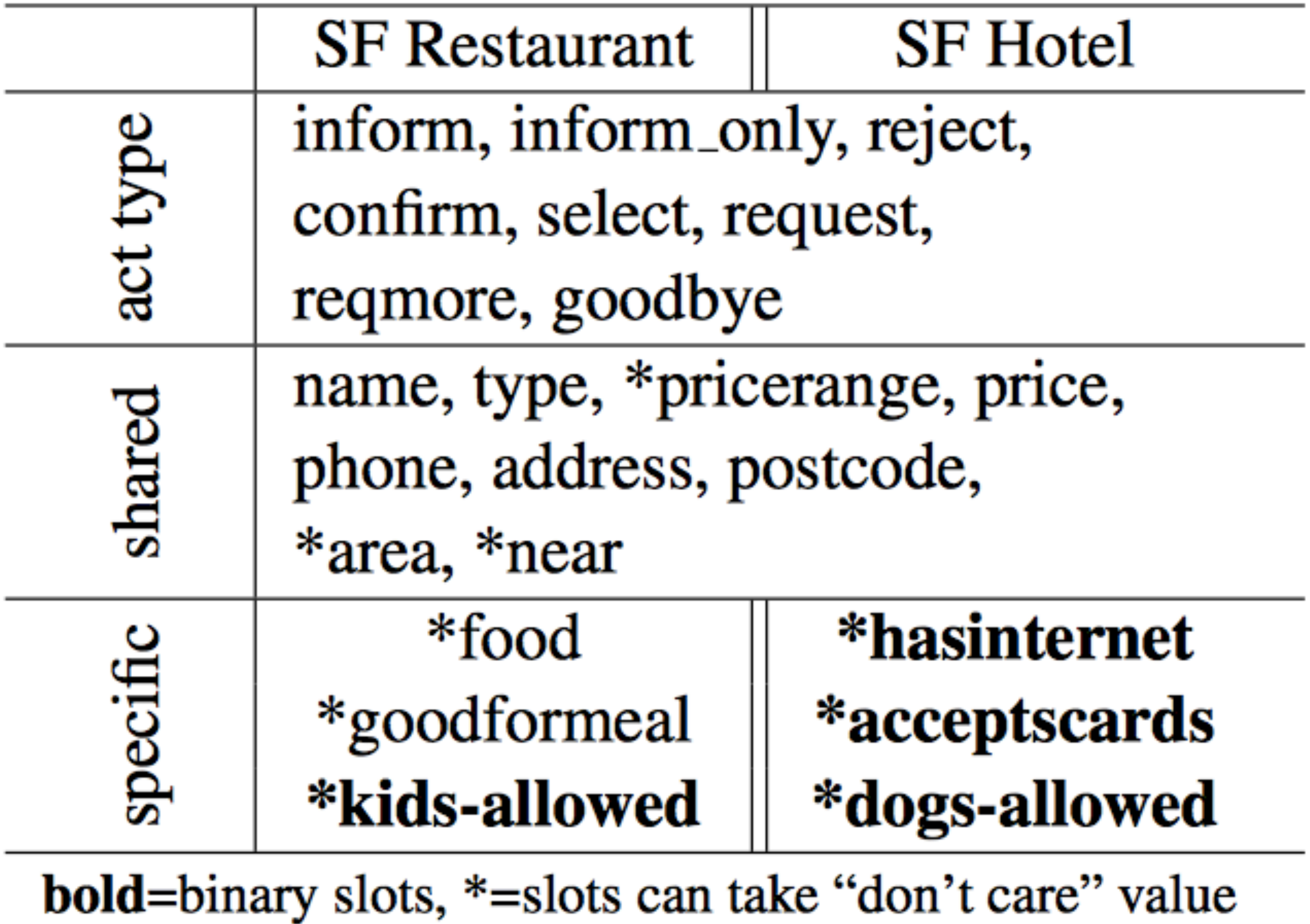
- e.g. a phrase-based translation model and a neural MT model (Niehues et al. 2017)

- Stacking uses the **output of one system in calculating features for another** system

Case Studies in Conditional Language Modeling

From Structured Data
(e.g. Wen et al 2015)

- When you say “Natural Language Generation” to an old-school NLPer, it means this

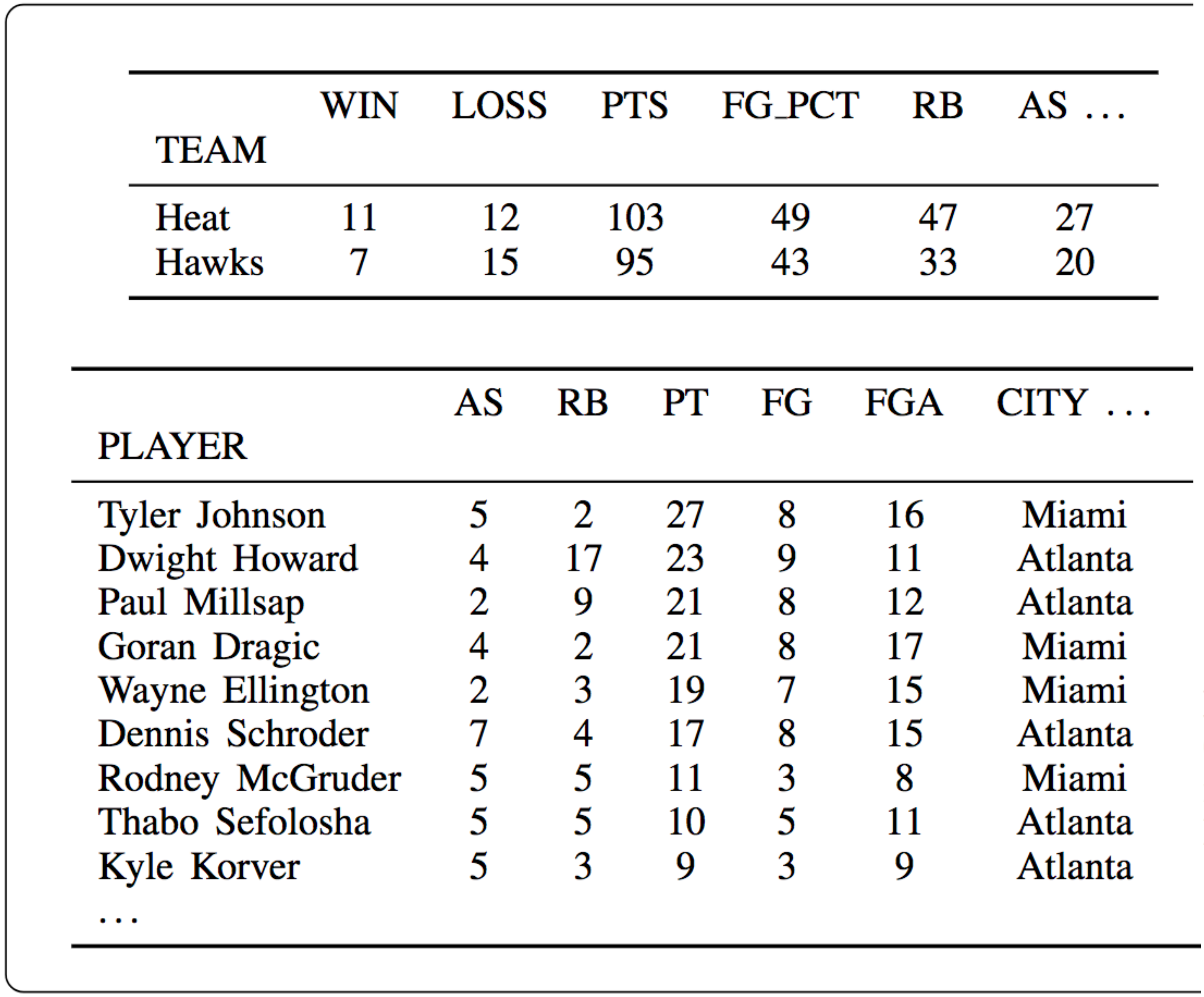


	SF Restaurant	SF Hotel
act type	inform, inform_only, reject, confirm, select, request, reqmore, goodbye	
shared	name, type, *pricerange, price, phone, address, postcode, *area, *near	
specific	*food *goodformeal ***kids-allowed**	***hasinternet** ***acceptscards** ***dogs-allowed**

bold=binary slots, *=slots can take “don’t care” value

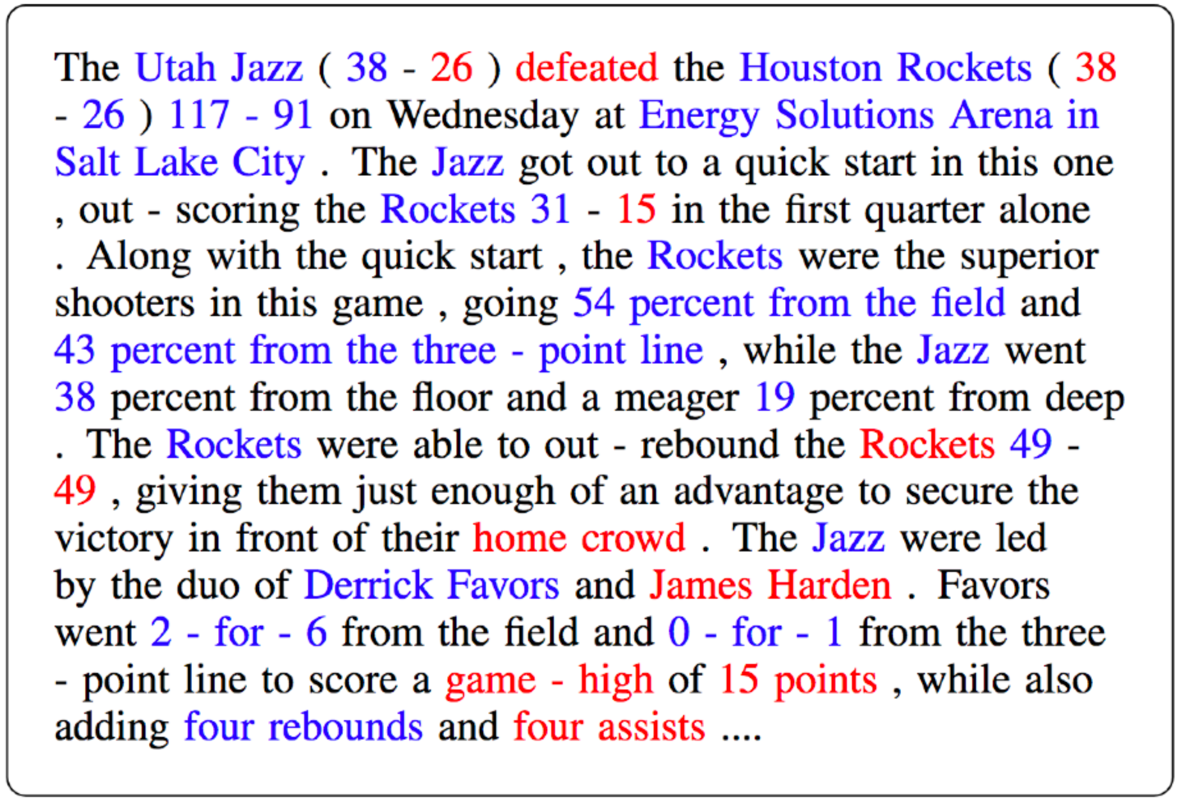
Still a Difficult Problem!

- e.g. "Challenges in data-to-document generation" (Wiseman et al. 2017)



TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS . . .
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY . . .
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
. . .						



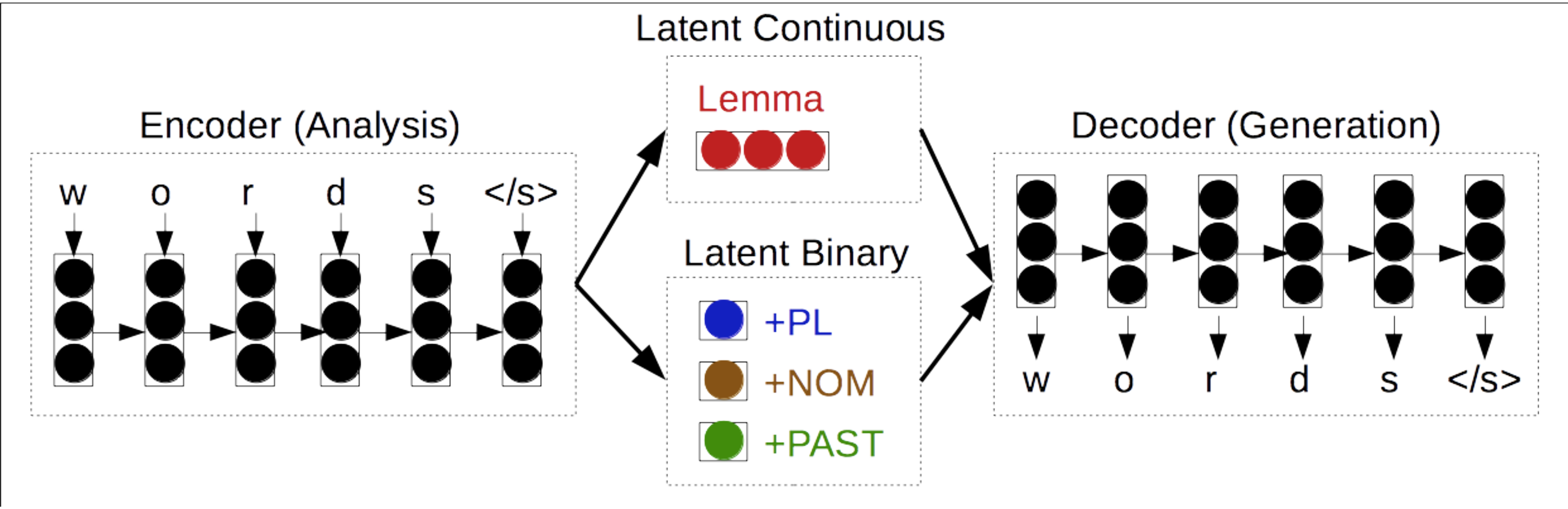
The Utah Jazz (38 - 26) defeated the Houston Rockets (38 - 26) 117 - 91 on Wednesday at Energy Solutions Arena in Salt Lake City . The Jazz got out to a quick start in this one , out - scoring the Rockets 31 - 15 in the first quarter alone . Along with the quick start , the Rockets were the superior shooters in this game , going 54 percent from the field and 43 percent from the three - point line , while the Jazz went 38 percent from the floor and a meager 19 percent from deep . The Rockets were able to out - rebound the Rockets 49 - 49 , giving them just enough of an advantage to secure the victory in front of their home crowd . The Jazz were led by the duo of Derrick Favors and James Harden . Favors went 2 - for - 6 from the field and 0 - for - 1 from the three - point line to score a game - high of 15 points , while also adding four rebounds and four assists ....

Figure 2: Example document generated by the Conditional Copy system with a beam of size 5. Text that accurately reflects a record in the associated box- or line-score is highlighted in blue, and erroneous text is highlighted in red.

- Focused evaluation using, e.g. information extraction

From Input + Labels
(e.g. Zhou and Neubig 2017)

- For example, word + morphological tags -> inflected word

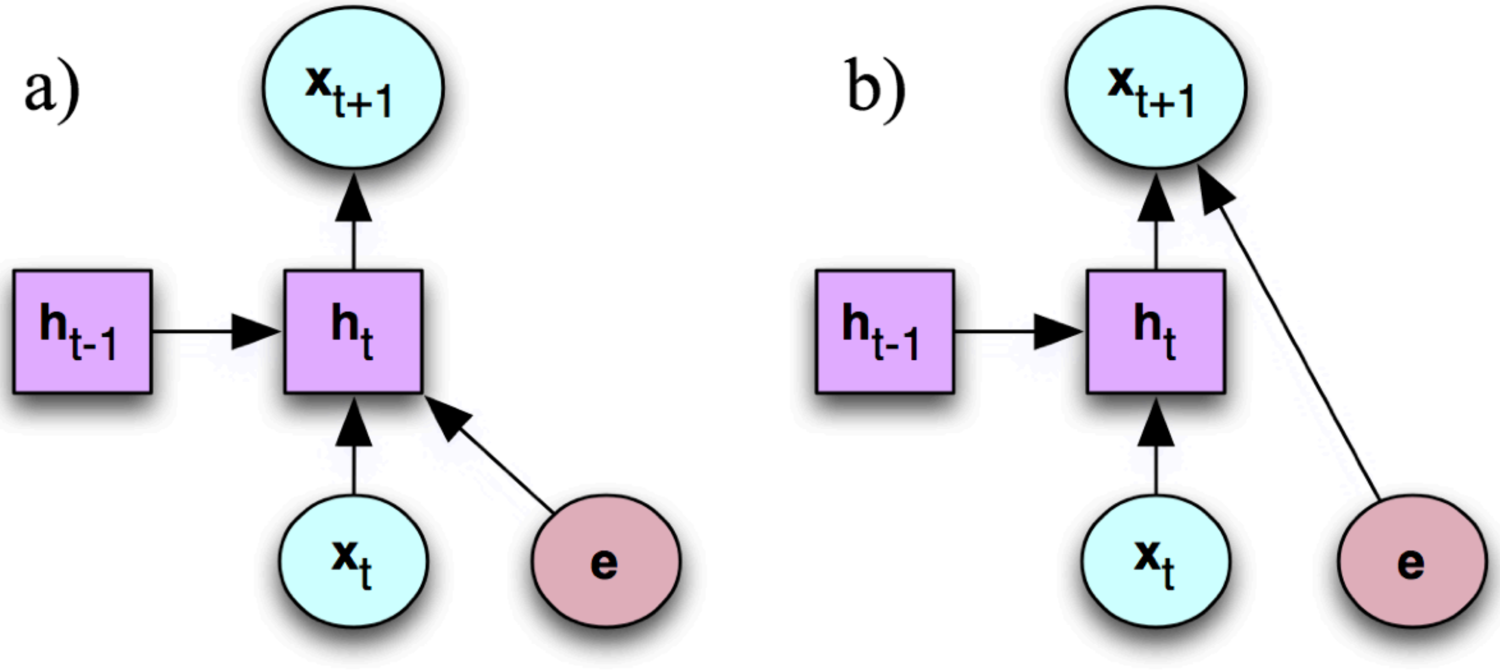


- Other options: politeness/gender in translation, etc.

From Speaker/Document Traits

(Hoang et al. 2016)

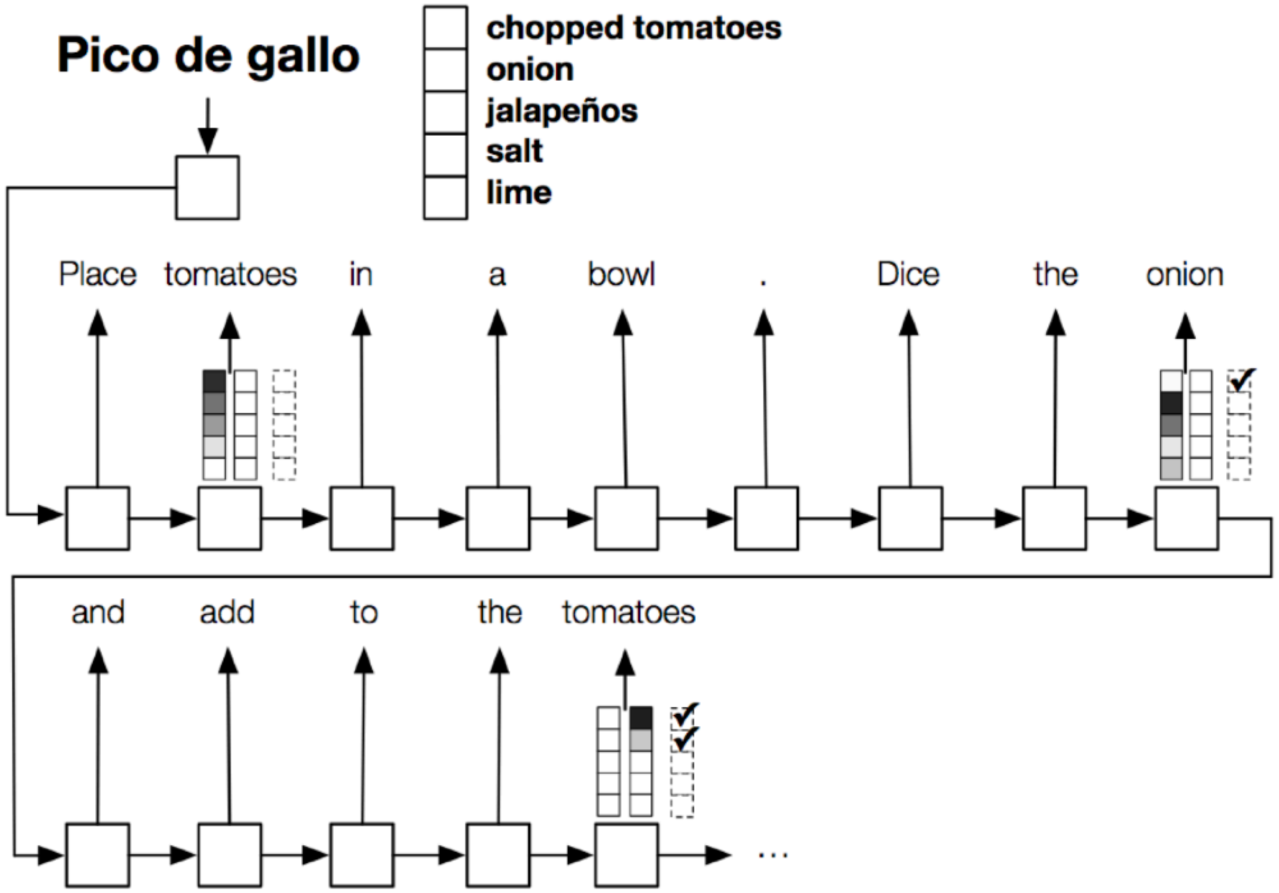
- e.g. TED talk description -> TED talk
- Encode title, description, keywords, author embedding
- Various encoding methods: BOW, CNN, RNN
- Various integration methods: in recurrent layer or softmax layer



From Lists of Traits
(Kiddon et al. 2016)

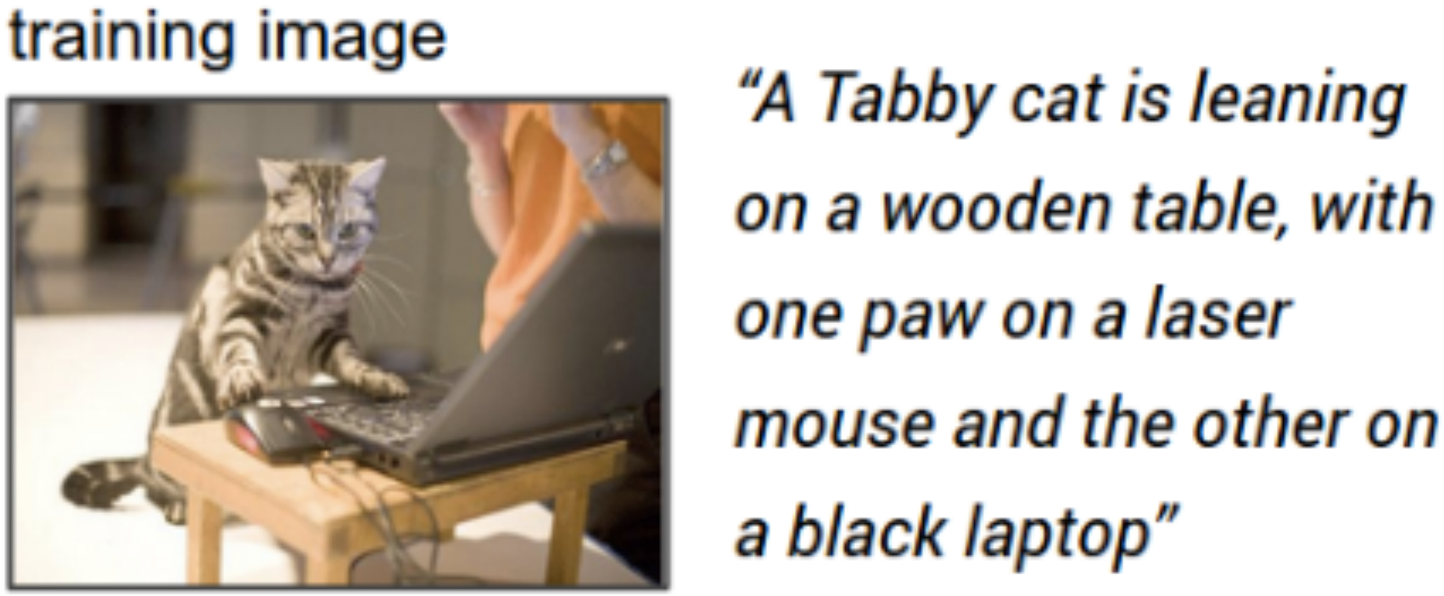
- Name of a recipe + ingredients -> recipe

- "Neural Checklist Model" that tells when a particular item in the list has been generated



From Images
(e.g. Karpathy et al. 2015)

- Input is image features, output is text

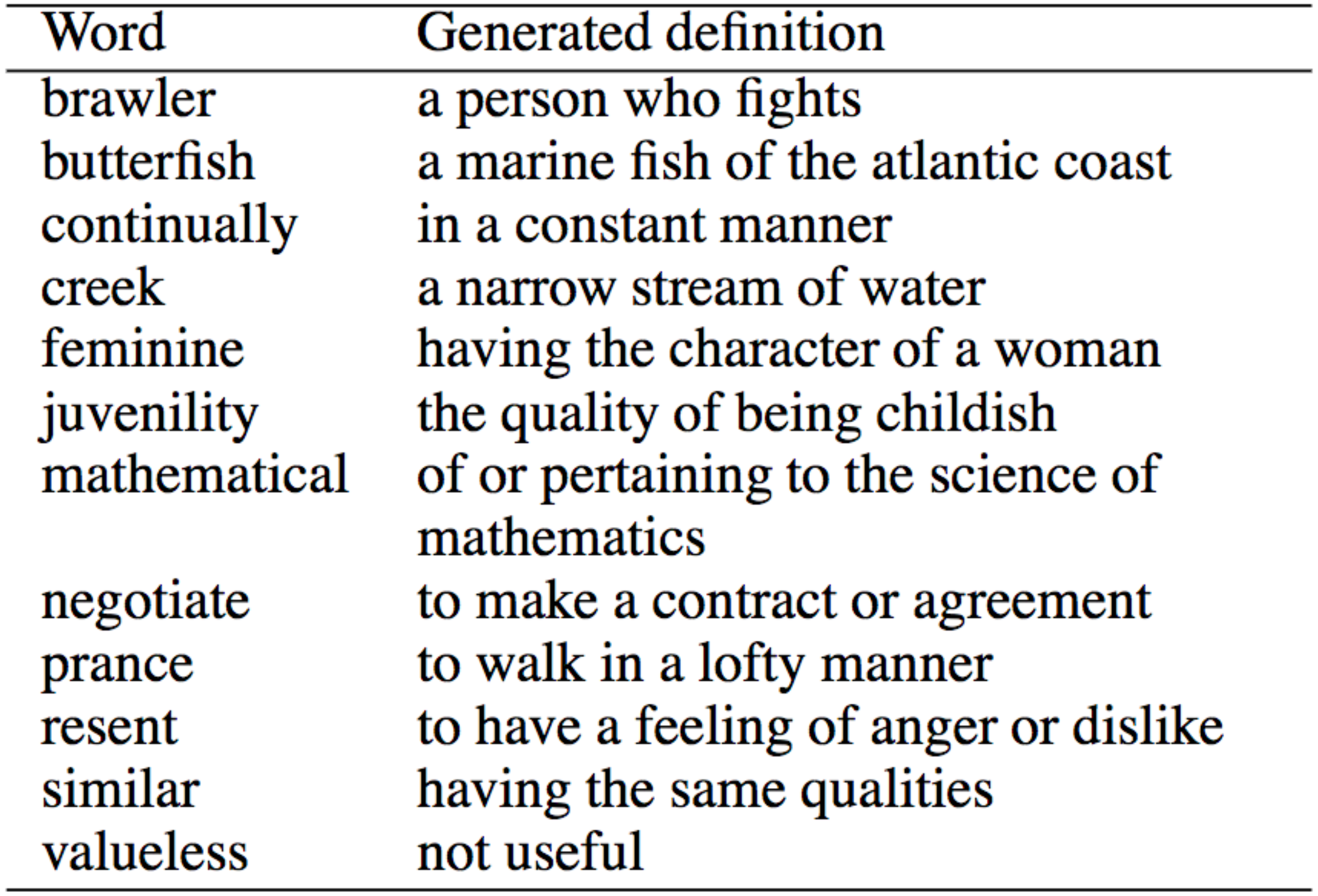


"A Tabby cat is leaning on a wooden table, with one paw on a laser mouse and the other on a black laptop"

- Use standard image encoders (e.g. CNN)
- Often pre-trained on large databases such as ImageNet

From Word Embeddings

(Noraset et al. 2017)



Word	Generated definition
brawler	a person who fights
butterfish	a marine fish of the atlantic coast
continually	in a constant manner
creek	a narrow stream of water
feminine	having the character of a woman
juvenility	the quality of being childish
mathematical	of or pertaining to the science of mathematics
negotiate	to make a contract or agreement
prance	to walk in a lofty manner
resent	to have a feeling of anger or dislike
similar	having the same qualities
valueless	not useful

- Baseline: standard sequence-to-sequence model
- Additional information about the affixes and hypernyms

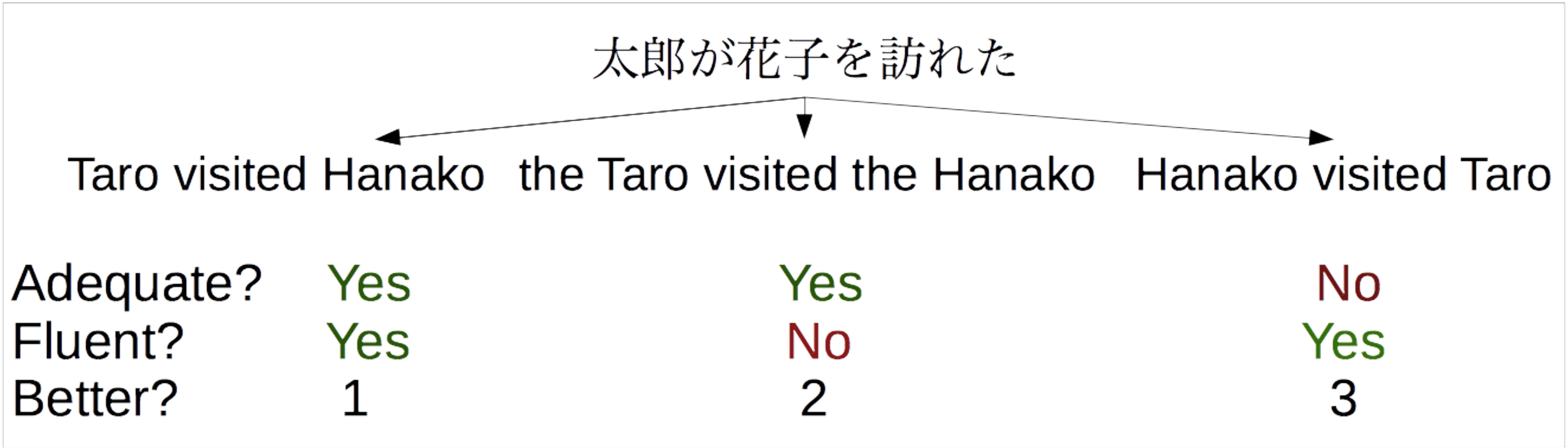
How do we Evaluate?

Basic Evaluation Paradigm

- Use parallel test set
- Use system to generate translations
- Compare target translations w/ reference

Human Evaluation

- Ask a human to do evaluation



太郎が花子を訪れた

	Taro visited Hanako	the Taro visited the Hanako	Hanako visited Taro
Adequate?	Yes	Yes	No
Fluent?	Yes	No	Yes
Better?	1	2	3

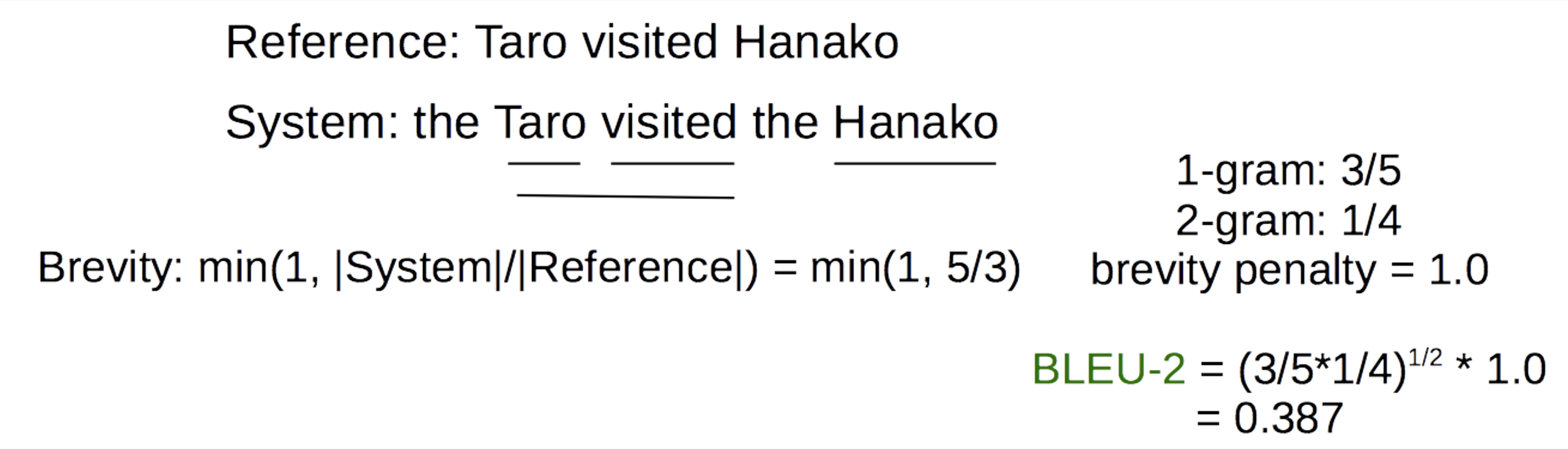
- Final goal, but slow, expensive, and sometimes inconsistent

Human Evaluation Shared Tasks

- **Machine Translation**
  - Conference on Machine Translation (WMT) shared tasks
    http://www.statmt.org/wmt20/
- **Composite Leaderboard**
  - GENIE leadeboard for QA, summarization, MT
    https://genie.apps.allenai.org/

BLEU

- Works by comparing n-gram overlap w/ reference



Reference: Taro visited Hanako

System: the Taro visited the Hanako

1-gram: 3/5
2-gram: 1/4
brevity penalty = 1.0

Brevity: min(1, |System|/|Reference|) = min(1, 5/3)

BLEU-2 = $(3/5*1/4)^{1/2} * 1.0$
= 0.387

- **Pros:** Easy to use, good for measuring system improvement
- **Cons:** Often doesn’t match human eval, bad for comparing very different systems

Embedding-based Metrics

- Recently, many metrics based on neural models
  - **BertScore:** Find similarity between BERT embeddings (unsupervised) (Zhang et al. 2020)
  - **BLEURT:** Train BERT to predict human evaluation scores (Sellam et al. 2020)
  - **COMET:** Train model to predict human eval, also using source sentence (Rei et al. 2020)
  - **PRISM:** Model based on training paraphrasing model (Thompson and Post 2020)

Perplexity

- Calculate the perplexity of the words in the held-out set *without* doing generation

- **Pros:** Naturally solves multiple-reference problem!

- **Cons:** Doesn’t consider decoding or actually generating output.

- May be reasonable for problems with lots of ambiguity.

Which One to Use?

- **Meta-evaluation** runs human evaluation and automatic evaluation on the same outputs, calculates correlation
- Examples:
  - **WMT Metrics Task** for MT (Mathur et al. 2021)
  - **RealSumm** for summarization ()
- Evaluation is hard, especially with good systems! Most metrics had no correlation w/ human eval over best systems at some WMT 2019 tasks

Questions?