COMP 790.139 (Fall 2017)
Natural Language Processing

Machine Translation 3 (Neural); Dialogue Models

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Mohit Bansal

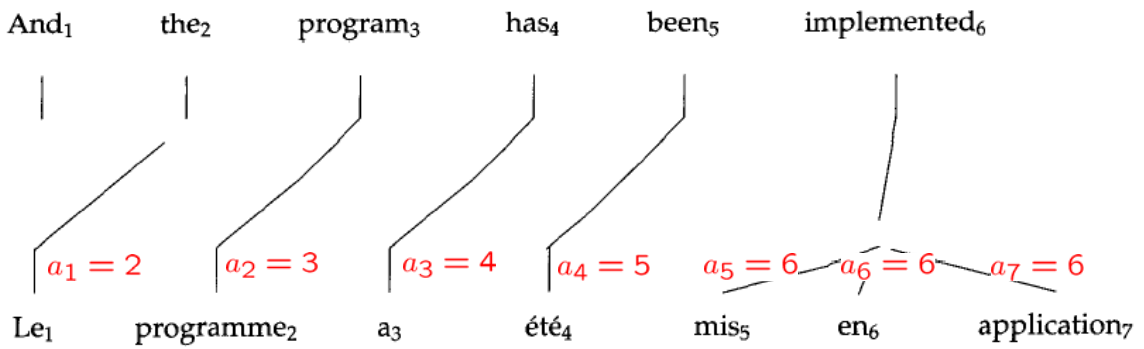
(various slides adapted/borrowed from courses by Dan Klein, JurafskyMartin-SLP3, Manning/Socher, others)

Statistical Machine Translation Recap

IBM Model 1

- Alignments: a hidden vector called an alignment specifies which English source is responsible for each French target word.
- The first, simplest IBM model treated alignment probabilities as roughly uniform:

$a = a_1 \ldots a_J$



$$P(f, a|e) = \prod_j P(a_j = i) P(f_j|e_i)$$

$$= \prod_j \frac{1}{I+1} P(f_j|e_i)$$

$$P(f|e) = \sum_a P(f, a|e)$$

[Brown et al., 1993]

IBM Model 2 (Distortion)

- The next more advanced model captures the notion of 'distortion', i.e., how far from the diagonal is the alignment

$$P(f, a|e) = \prod_j P(a_j = i|j, I, J) P(f_j|e_i)$$

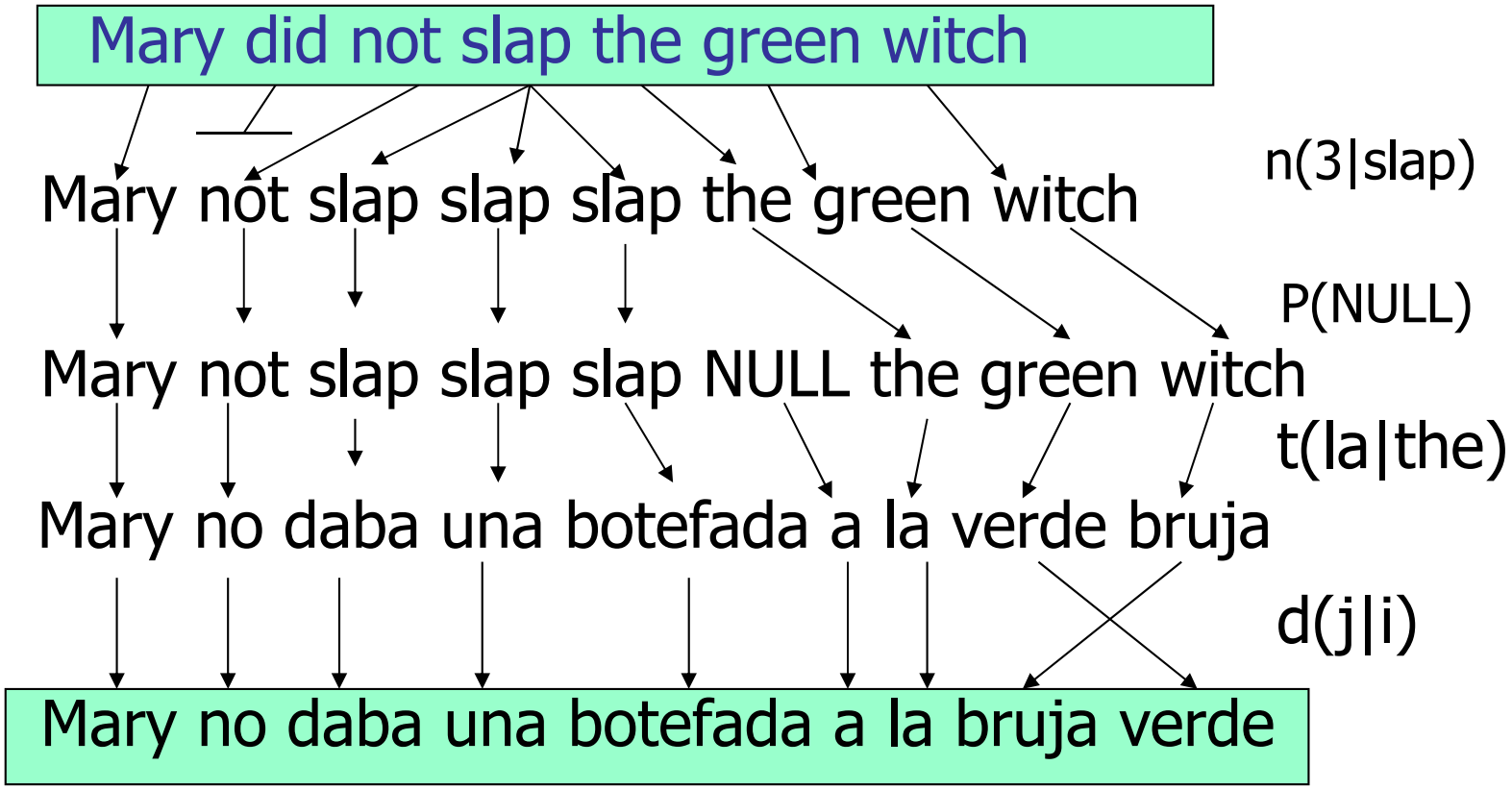
$$P(dist = i - j\frac{I}{J})$$

$$\frac{1}{Z} e^{-\alpha(i - j\frac{I}{J})}$$

- Other approaches for biasing alignment towards diagonal include relative vs absolute alignment, asymmetric distances, and learning a full multinomial over distances

[Brown et al., 1993]

IBM Models 3/4/5 (Fertility)



[Vogel et al., 1996]

Synchronous Tree-Substitution Grammars

STSG extraction

1. Phrases
   - respect word alignments
   - are syntactic constituents on *both* sides
2. Phrase pairs form rules
3. Subtract phrases to form rules

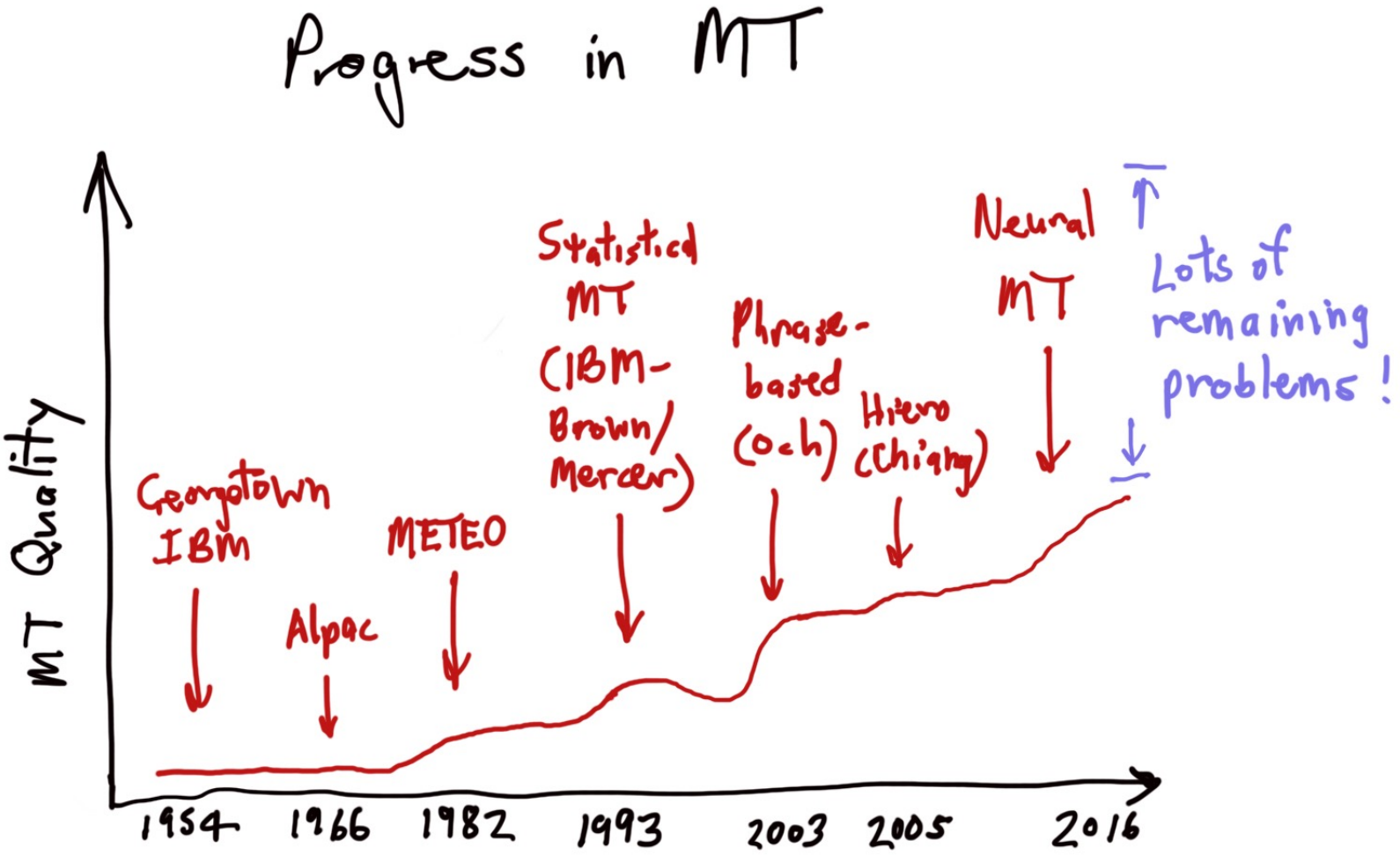
[Shieber, 2004; Graehl et al., 2008]

Neural Machine Translation

Traditional Stat. Machine Translation

- Lots of feature engineering
- Very complex pipeline systems with multiple steps to generate the final huge phrase table!
- Incentive to do it end-to-end and jointly
- Can neural models be a powerful enough alternative to do so?

Machine Translation Progress

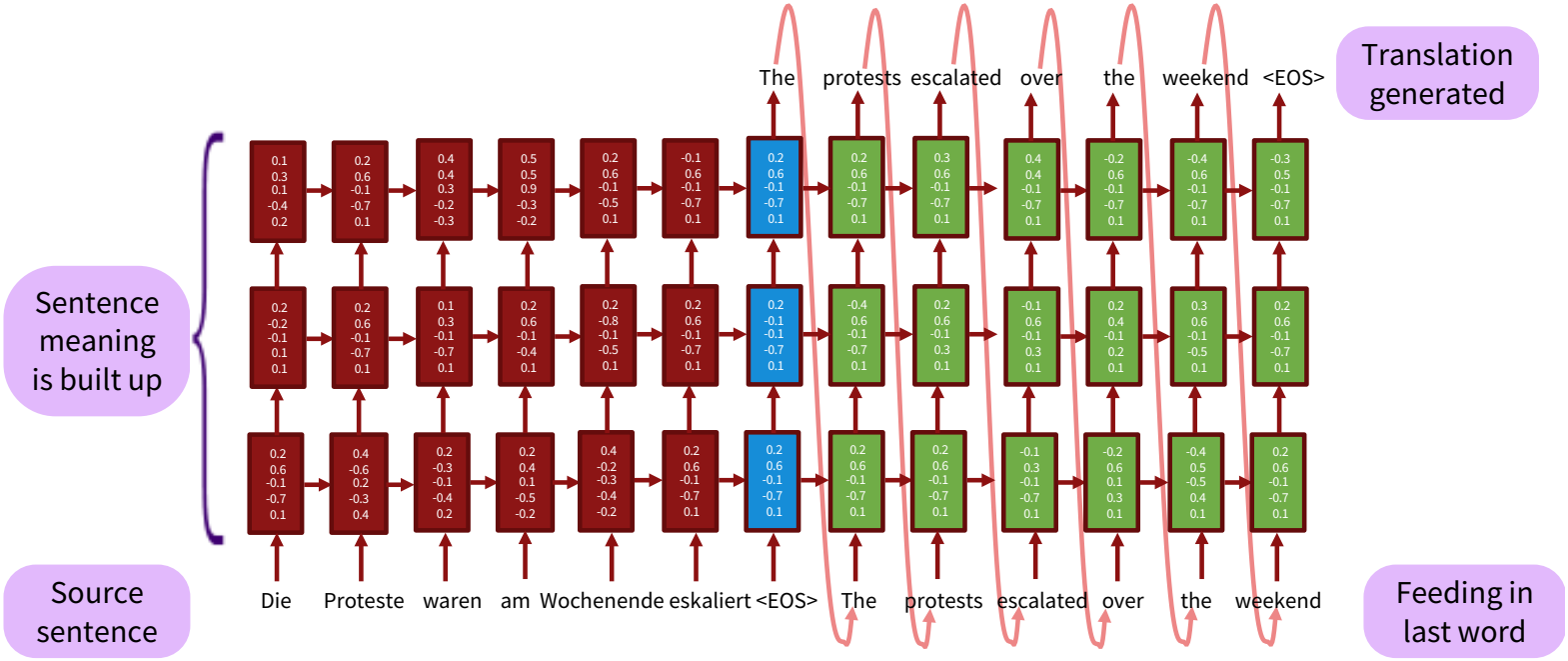


NMT slides from ACL 2016 Tutorial (Luong, Cho, Manning)

Neural Machine Translation

- Encoder-Decoder RNN models:

[Sutskever et al. 2014, Bahdanau et al. 2014, et seq.]
following [Jordan 1986] and more closely [Elman 1990]



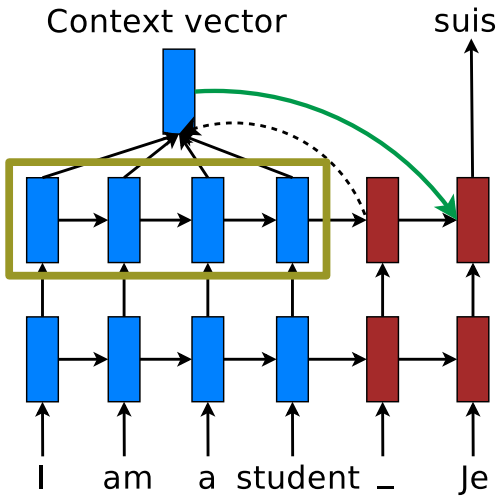
A deep recurrent neural network

Initial Improvement Sources

- Stacking multiple layers
- Bidirectionality
- Better memory units, e.g., GRUs
- Pre-trained language models on tons of monolingual data
- Ensembles
- Attention/Alignment models

Alignment/Attention Models

- Translating longer sentences better, e.g., via attention/alignment module between encoder and decoder to jointly learn alignments and translations end-to-end

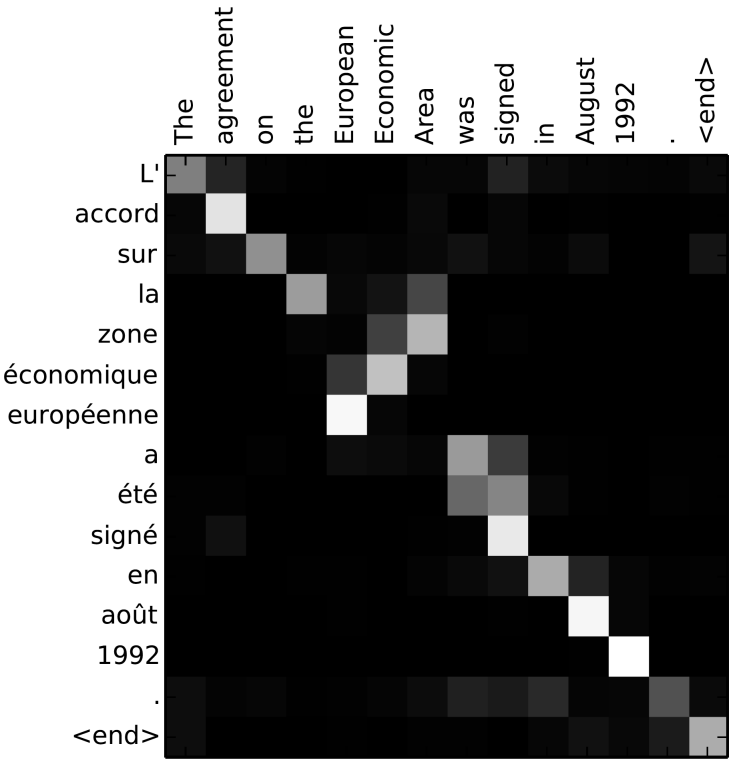


$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) \end{cases}$$

Bilinear form: well-adopted.

Alignment/Attention Models

- Translating longer sentences better, e.g., via attention/alignment module between encoder and decoder to jointly learn alignments and translations end-to-end

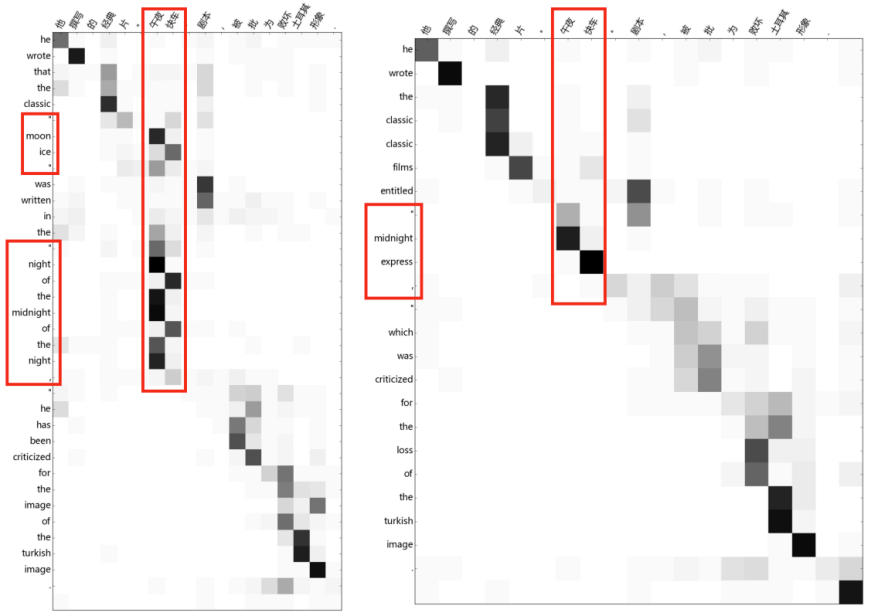


Dzmitry Bahdanau, KyungHuyn Cho, and Yoshua Bengio. ***Neural Machine Translation by Jointly Learning to Translate and Align****. ICLR'15.*

Linguistic Insights in NMT

Constraints on "distortion" (displacement) and fertility

➜ **Constraints on attention** [Cohn, Hoang, Vymolova, Yao, Dyer & Haffari NAACL 2016; Feng, Liu, Li, Zhou 2016 arXiv; Yang, Hu, Deng, Dyer, Smola 2016 arXiv].



Linguistic Insights in NMT

Extend to NMT – *Linguistic insights*

- [Cohn, Hoang, Vymolova, Yao, Dyer, Haffari, NAACL'16]: position (IBM2) + Markov (HMM) + fertility (IBM3-5) + alignment symmetry (BerkeleyAligner).

$$-\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2$$

Per source word — Source word fertility

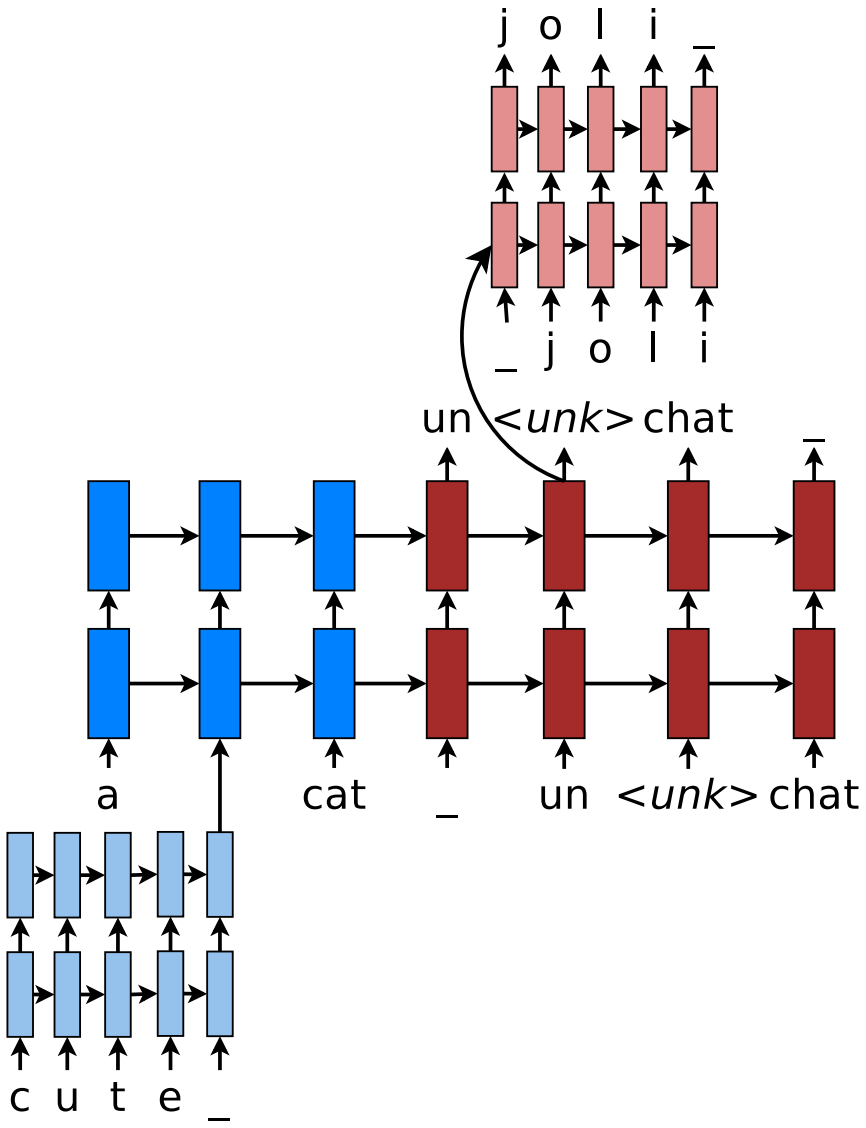
- [Tu, Lu, Liu, Liu, Li, ACL'16]: linguistic & NN-based coverage models.

Other New Ideas/Improvements

- Extending vocabulary coverage and handling rare/unseen words
- Handling more language variations, e.g., via character-level models to capture morphology
- Utilize more data resources, e.g., multilingual models (one to many, many to one, many to many), multi-task learning (combine with other encoder-decoder tasks with shared sides)
- Zero-shot translation

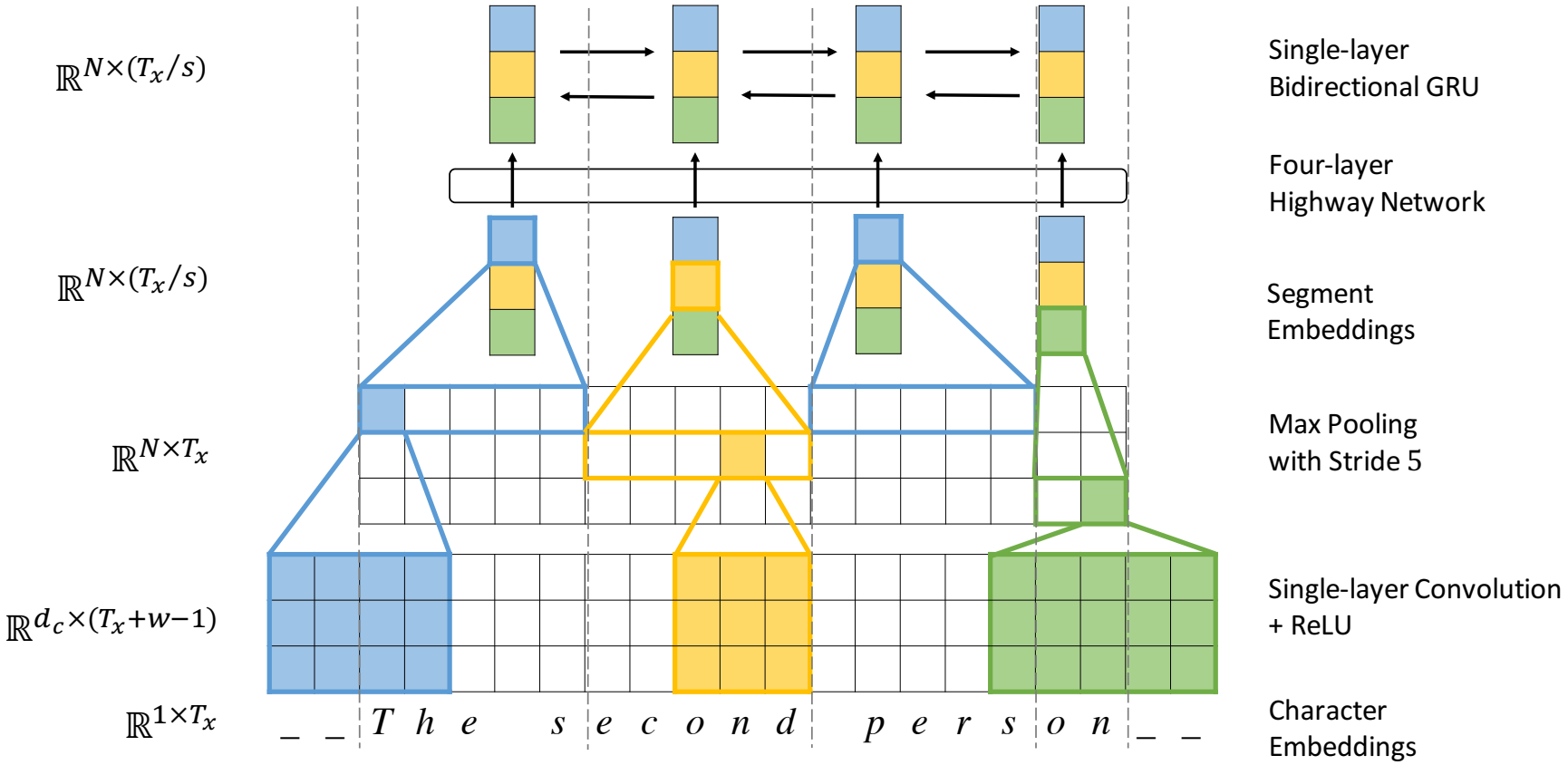
- See ACL 2016 tutorial: https://sites.google.com/site/acl16nmt/

Hybrid Char-Word NMT



[Luong and Manning, 2016]

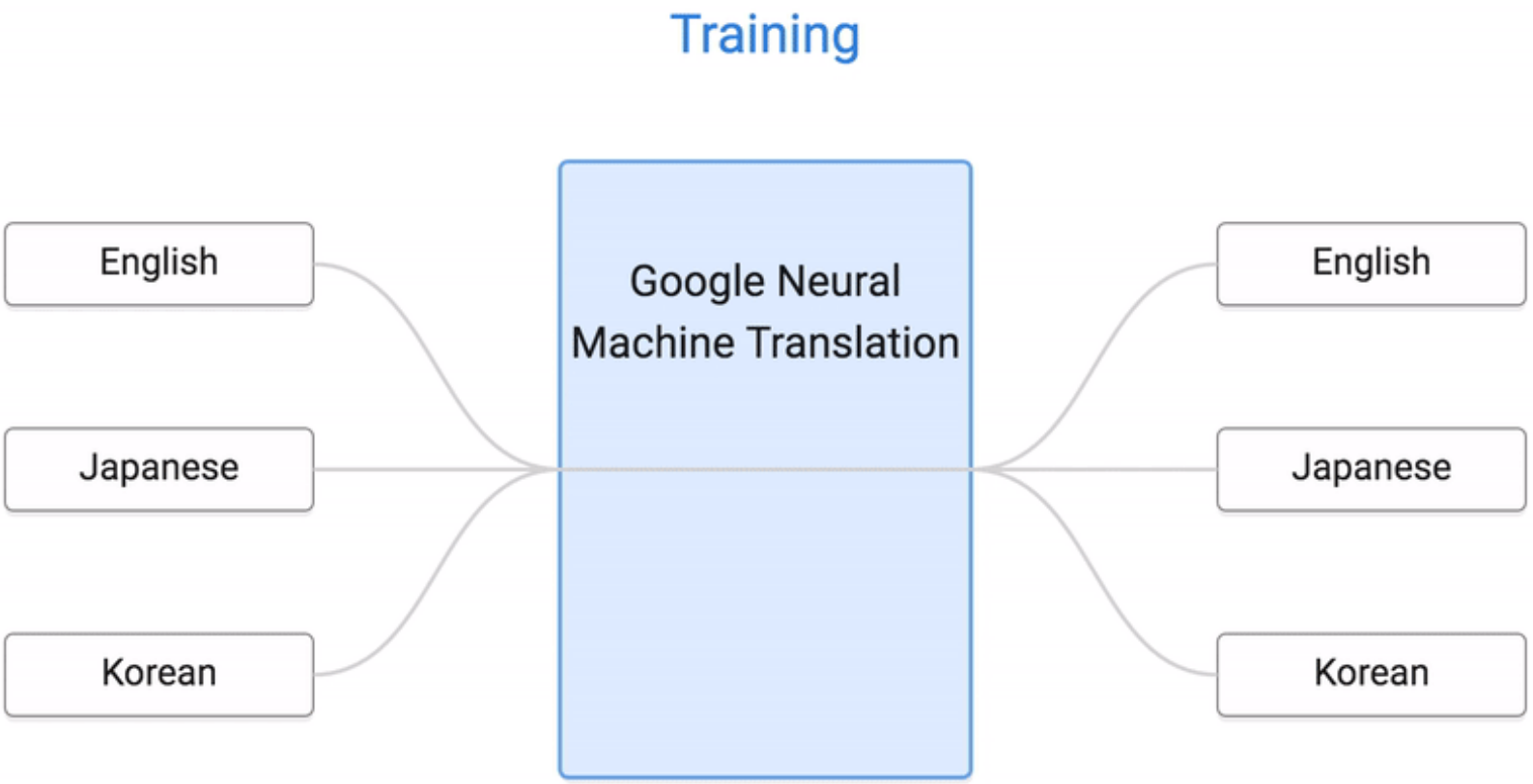
Char-level NMT with CNN Encoder



- Later extended to convolutions for both encoder and decoder!

[Lee et al., 2016; Gehring et al., 2016]

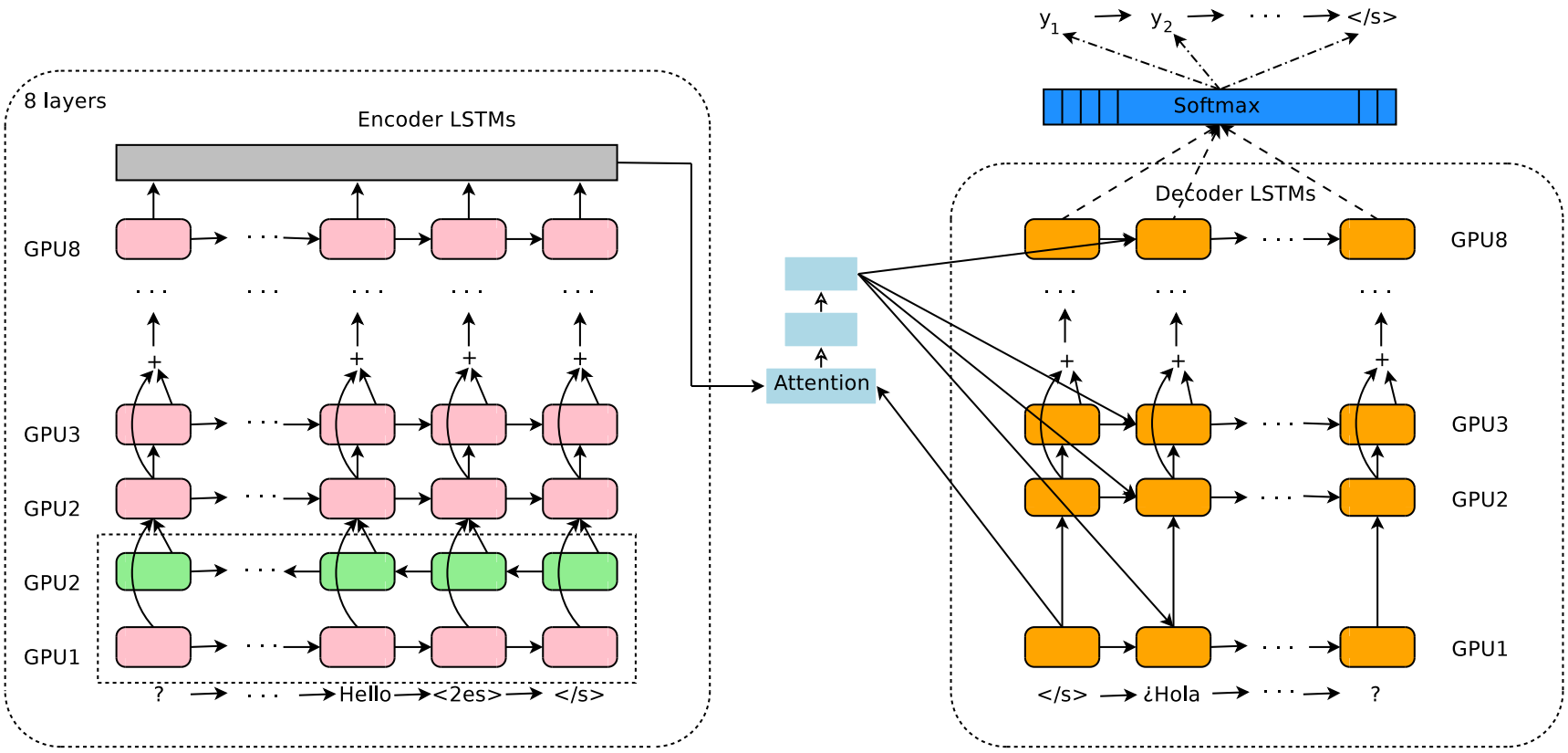
Google's Zero-Shot Machine Translation



- Play above gif video at https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html

[Johnson et al., 2016]

Google's Zero-Shot Machine Translation



[Johnson et al., 2016]

Dialogue Models

Examples

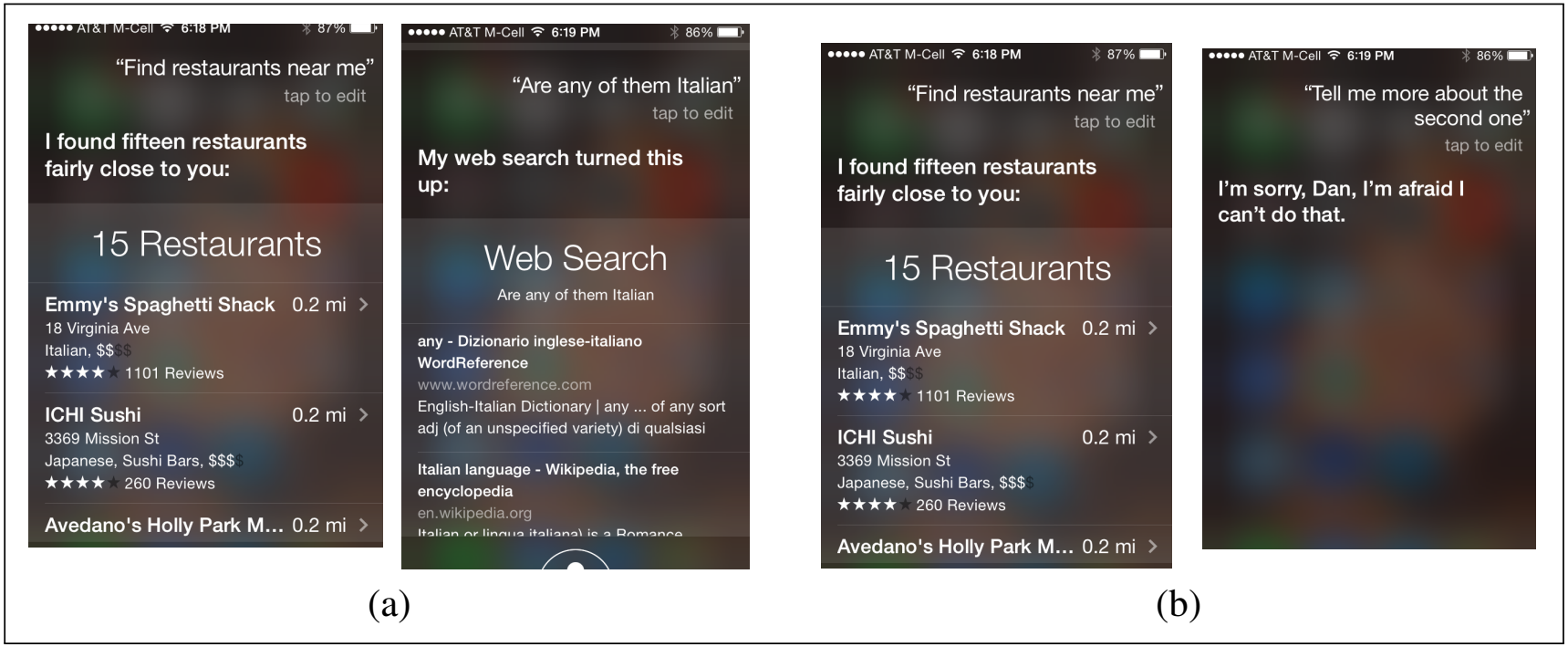


Figure 29.1 Two sets of interactions with Siri in 2014. (a) A question (”Find restaurants near me”) returns restaurants, but the system was unable to interpret a follow-up question (“Are any of them Italian?”). (b) An alternative followup (“Tell me more about the second one”) similarly fails. This early system’s confusion at follow-up questions suggests that it is mainly designed for a single interaction.

[Jurafsky SLP3 Book Chapter 29]

Examples

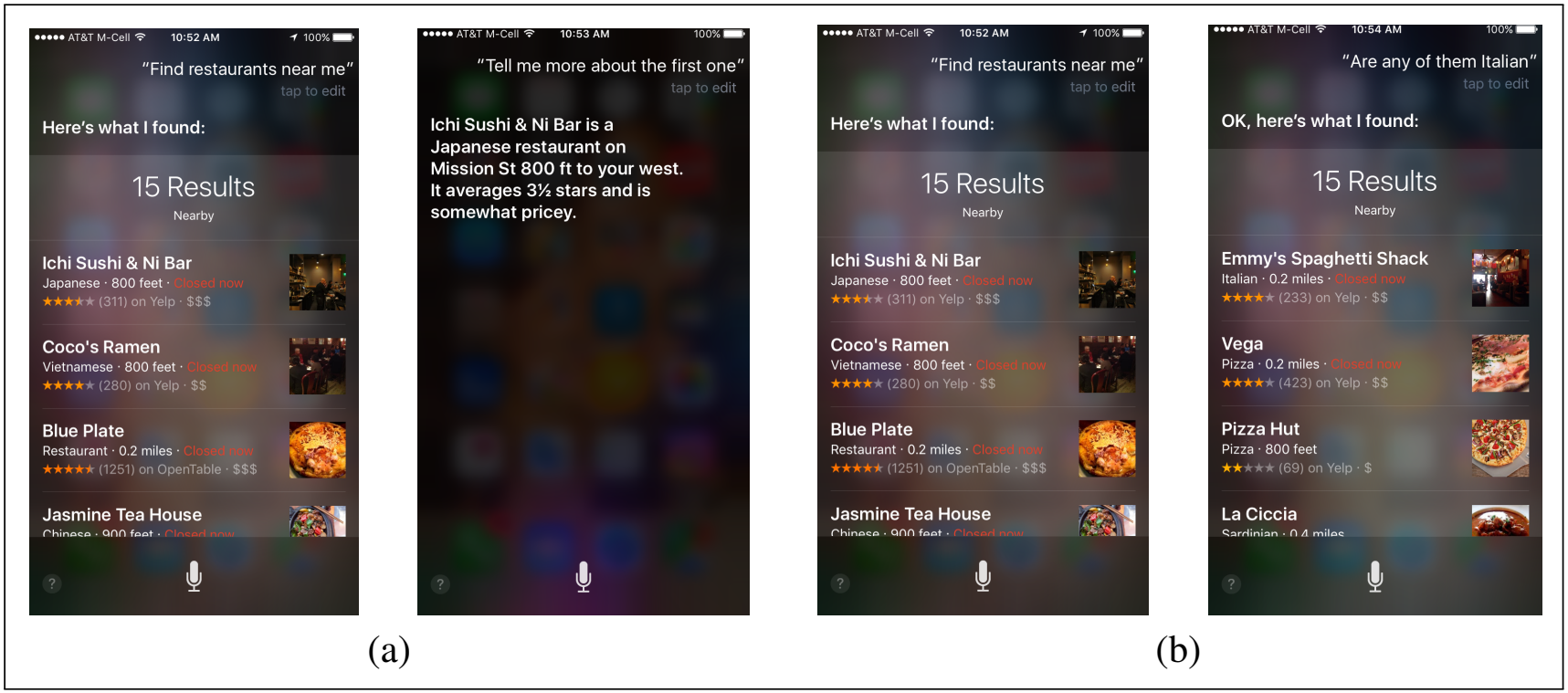


Figure 29.2 The same follow-up questions that Siri couldn't answer in 2014 receive appropriate responses when posed to Siri in 2017.

Examples

Tutor:	What will the velocity of the object be a second after that (where the initial velocity is 9.8 m/s and the acceleration is 9.8 m/s^2)?
Student:	Is it 19.6 m/s?
Tutor:	Good. A second later, its velocity will be 9.8m/s + 9/8 m/s = 19.6 m/s This is because its acceleration tells us that every second, its velocity increases by 9.8 m/s. So what'll its velocity be, a second after that?
Student:	Um. I'm thinking. 29.4 m/s
Tutor:	Right. Now back to the man and his keys. We've shown that they both have an acceleration of 9.8 m/s^s. So, how do their velocities compare with each other, at every second during the fall?

Figure 29.3 The tutoring domain: An excerpt from a tutorial interaction between a physics student and the ITSPOKE system of Forbes-Riley and Litman (2011) based on the Why2-Atlas text-based tutoring system (VanLehn et al., 2002).

Examples

User:	Do you like Doctor Who?
Cleverbot:	Yes, so funny.
User:	What's your favorite show?
Cleverbot:	I don't have one.
User:	What's your name?
Cleverbot:	Jason I already told you.
User:	No your real name.
Cleverbot:	Stephanie IS my real name.
User:	I thought it was Jason?
Cleverbot:	How did you come to that conclusion.
User:	Because you said so.
Cleverbot:	I have to do my homework.

Figure 29.4 Chatbots: A conversation between a user and Cleverbot, an IR-based chatbot (Carpenter, 2017).

Rule-based Chatbots (e.g., ELIZA)

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
..
..
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?
My father
YOUR FATHER
You are like my father in some ways.
WHAT RESEMBLANCE DO YOU SEE
You are not very aggressive but I think you don't want me to notice that.
WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
You don't argue with me.
WHY DO YOU THINK I DON'T ARGUE WITH YOU
You are afraid of me.
DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU
My father is afraid of everybody.
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
Bullies.
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

Rule-based Chatbots (e.g., ELIZA)

```
(0 YOU 0 ME) [pattern]
->
(WHAT MAKES YOU THINK I 3 YOU) [transform]
```

function ELIZA GENERATOR(user *sentence*) **returns** *response*

Find the word *w* in *sentence* that has the highest keyword rank
  if *w* exists
    Choose the highest ranked rule *r* for *w* that matches *sentence*
    response ← Apply the transform in *r* to *sentence*
    if *w* = 'my'
      future ← Apply a transformation from the 'memory' rule list to *sentence*
      Push *future* onto memory stack
  else (no keyword applies)
    either
      response ← Apply the transform for the NONE keyword to *sentence*
    or
      response ← Pop the top response from the memory stack
  return(*response*)

Figure 29.5 A simplified sketch of the ELIZA algorithm. The power of the algorithm come from the particular transforms associated with each keyword.

IR-based Chatbots

- Retrieval systems use two major approaches to "extract" the best response from a dialogue corpus, given the new, test-time user utterance:

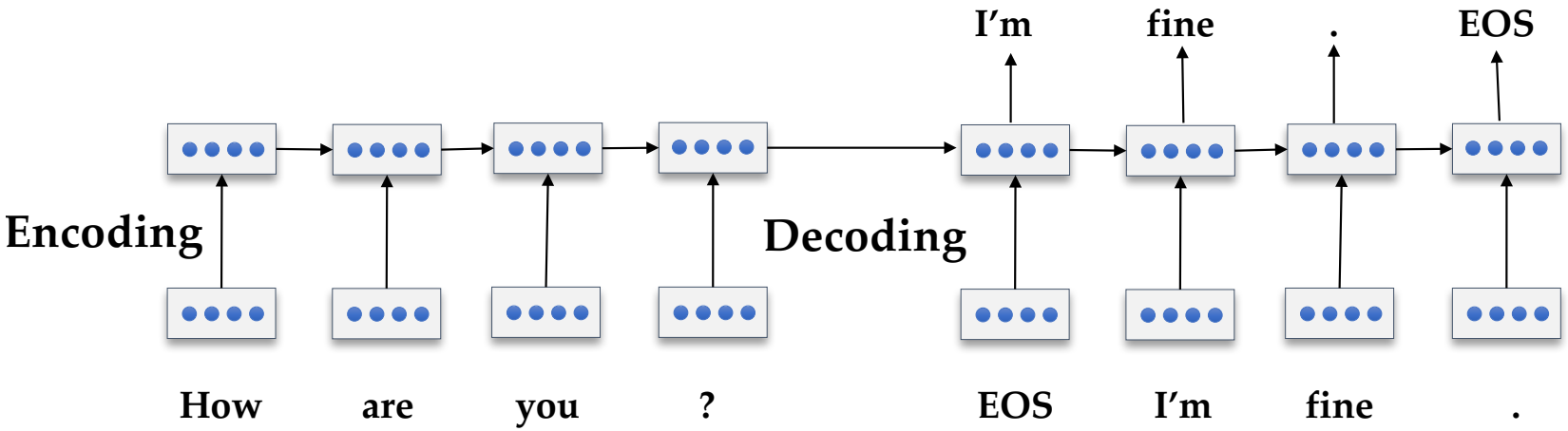
- **1) Return Response of Most Similar Turn:** Find conversation turn *t* (in corpus *C*) which is most similar to the given user utterance/query *q*, and return the following turn/response *r* of that most-similar utterance:

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

- **2) Return Most Similar Turn:** Instead of returning the following turn of the most similar utterance, we return this most similar utterance itself, with the intuition that a good response often shared words/semantics with the prior turn:

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

Seq-to-Seq Chatbots



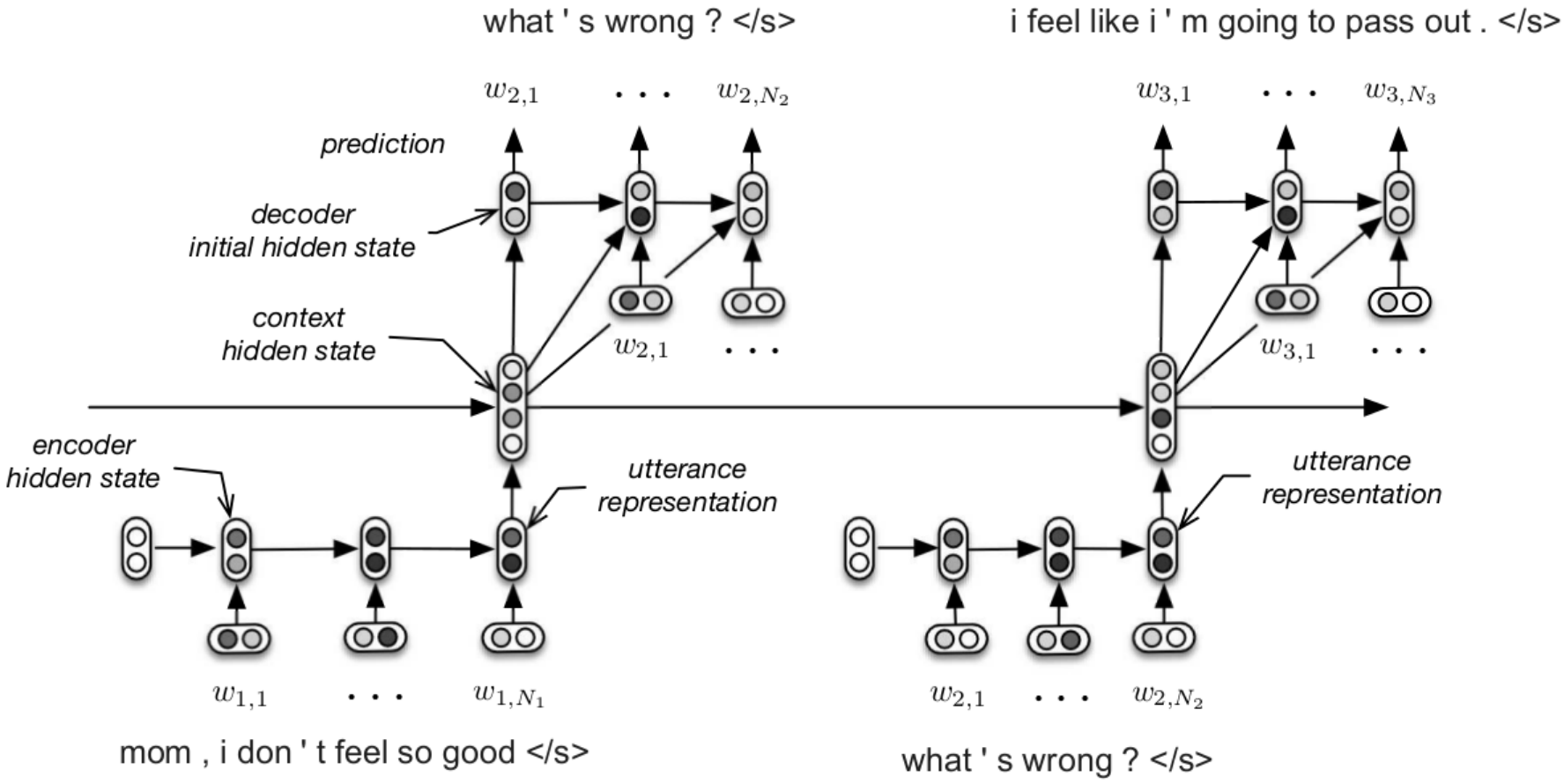
[Shang et al. 2015; Vinyals and Le, 2015; Sordoni et al., 2015]

Evaluating Chatbots

- Automatic metrics based on word/phrase overlap not very useful because so many responses might be correct/appropriate for chitchat
- Human evaluation most meaningful/common (but time-consuming)
- Can't do slot-filling techniques because this is not task-oriented dialogue with a specific goal or success metric
- Engagement or length of conversation in real human-based setup?
- Some new automatic classification approaches like ADEM [Lowe et al., 2017] to classify appropriateness of response, and Adversarial evaluation [Bowman et al., 2016; Kannan and Vinyals, 2016; Li et al., 2017] to fool a classifier that distinguishes between human and machine generated responses

Some Advanced Seq-to-Seq Models

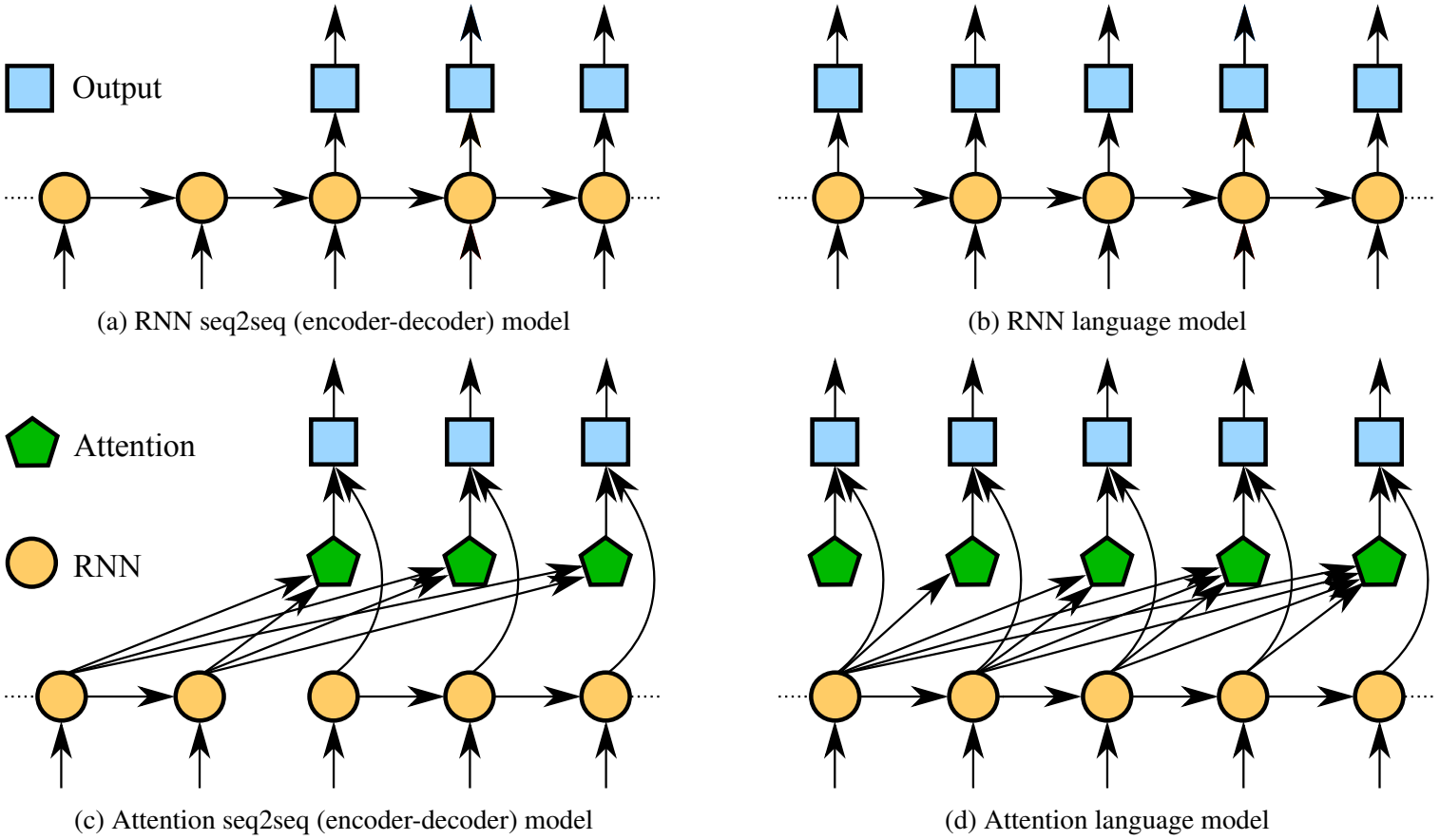
- Hierarchical Recurrent Encoder-Decoder



[Serban et al., 2015]

Some Advanced Seq-to-Seq Models

- Attention-RNN Language Model



(a) RNN seq2seq (encoder-decoder) model

(b) RNN language model

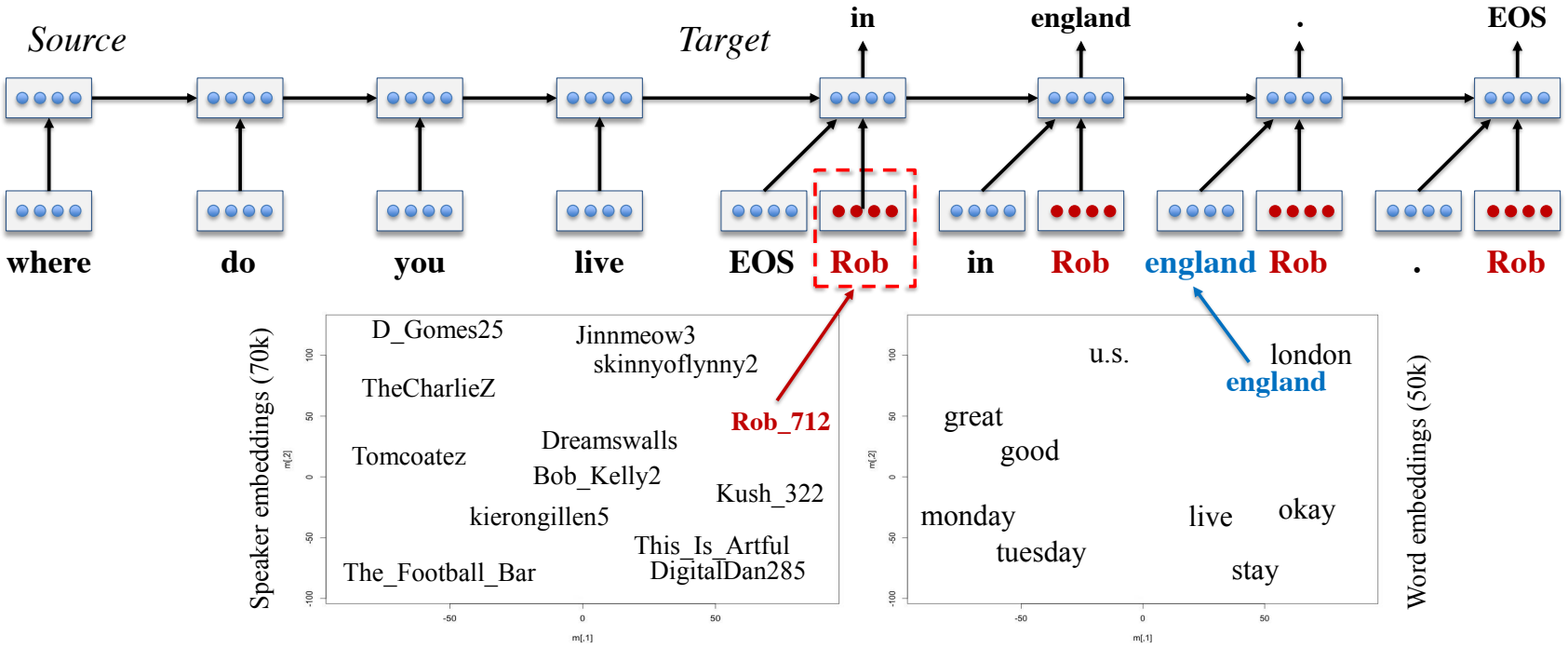
(c) Attention seq2seq (encoder-decoder) model

(d) Attention language model

[Mei et al., 2016]

Some Advanced Seq-to-Seq Models

- Persona-based Language Models



[Li et al., 2016]

Frame-based & KB-based Dialog Agents

GUS:	Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?
Client:	I want to go to San Diego on May 28
GUS:	What time do you want to leave?
Client:	I must be in San Diego before 10 am
GUS:	Would you like P.S.A. flight 102 that arrives at 9.15 am ?
Client:	What is the next flight ?
GUS:	Air California flight 310 that leaves at 8.30am and arrives at 10.00am
Client:	I'll take the first one
GUS:	In what name should I make a reservation, please?
Client:	Dan Bobrow
GUS:	I have confirmed the following flight:
Client:	P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.
GUS:	What date do you want to return on ?
Client:	On Friday in the evening.
GUS:	Would you like the flight that leaves at 7.45 pm ?
Client:	That's fine.
GUS:	I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

Figure 29.8 The travel domain: A transcript of an actual dialog with the GUS system of Bobrow et al. (1977). P.S.A. and Air California were airlines of that period.

[Jurafsky SLP3]

Frame-based & KB-based Dialog Agents

Slot	Type
ORIGIN CITY	city
DESTINATION CITY	city
DEPARTURE TIME	time
DEPARTURE DATE	date
ARRIVAL TIME	time
ARRIVAL DATE	date

```
DATE
  MONTH NAME
  DAY (BOUNDED-INTEGER 1 31)
  YEAR INTEGER
  WEEKDAY (MEMBER (SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY)]
```

Slot	Question
ORIGIN CITY	“From what city are you leaving?”
DESTINATION CITY	“Where are you going?”
DEPARTURE TIME	“When would you like to leave?”
ARRIVAL TIME	“When do you want to arrive?”

Frame-based & KB-based Dialog Agents

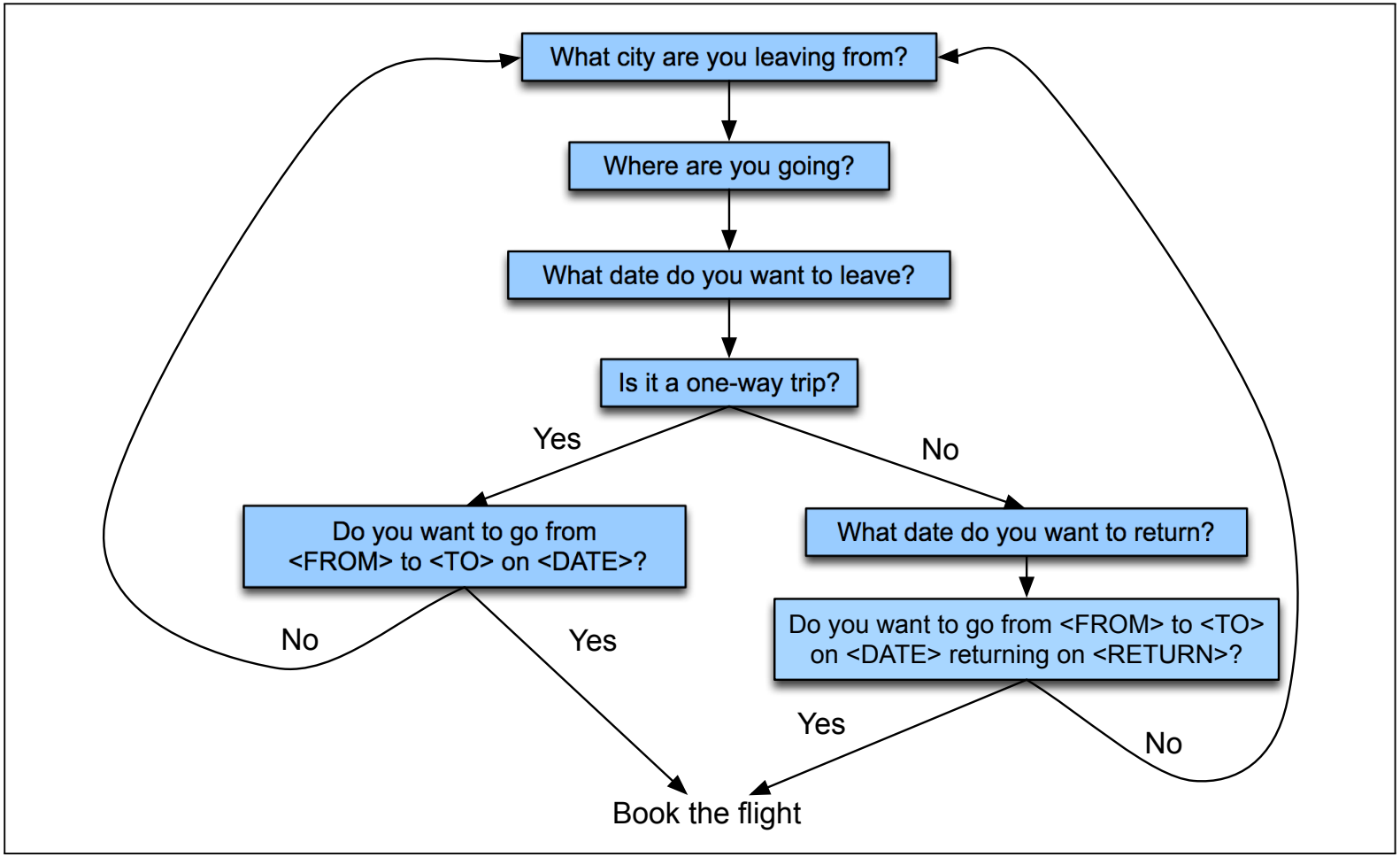


Figure 29.9 A simple finite-state automaton architecture for frame-based dialog.

Frame-based & KB-based Dialog Agents

`Show me morning flights from Boston to San Francisco on Tuesday`

a system might want to build a representation like:

```
DOMAIN:       AIR-TRAVEL
INTENT:       SHOW-FLIGHTS
ORIGIN-CITY:  Boston
ORIGIN-DATE:  Tuesday
ORIGIN-TIME:  morning
DEST-CITY:    San Francisco
```

while an utterance like

`Wake me tomorrow at 6`

should give an intent like this:

```
DOMAIN:  ALARM-CLOCK
INTENT:  SET-ALARM
TIME:    2017-07-01 0600-0800
```

Frame-based & KB-based Dialog Agents

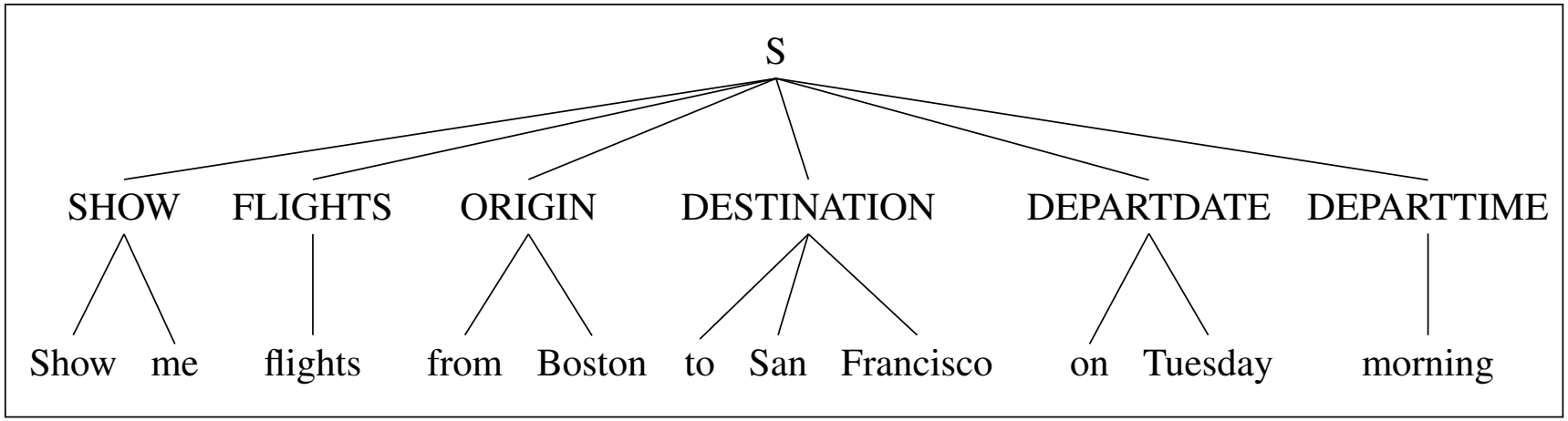


Figure 29.10 A semantic grammar parse for a user sentence, using slot names as the internal parse tree nodes.

Frame-based & KB-based Dialog Agents

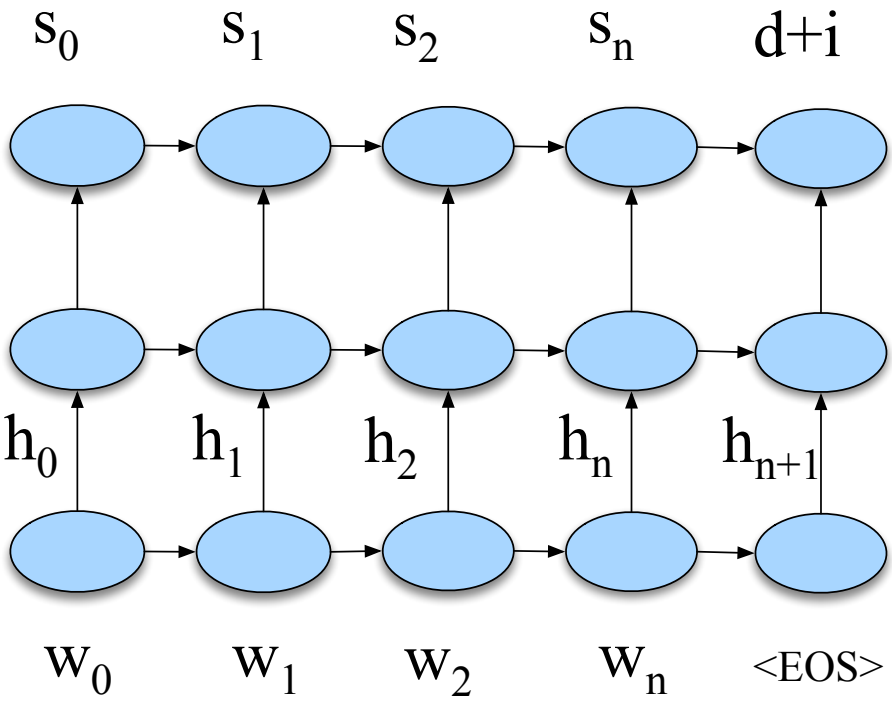


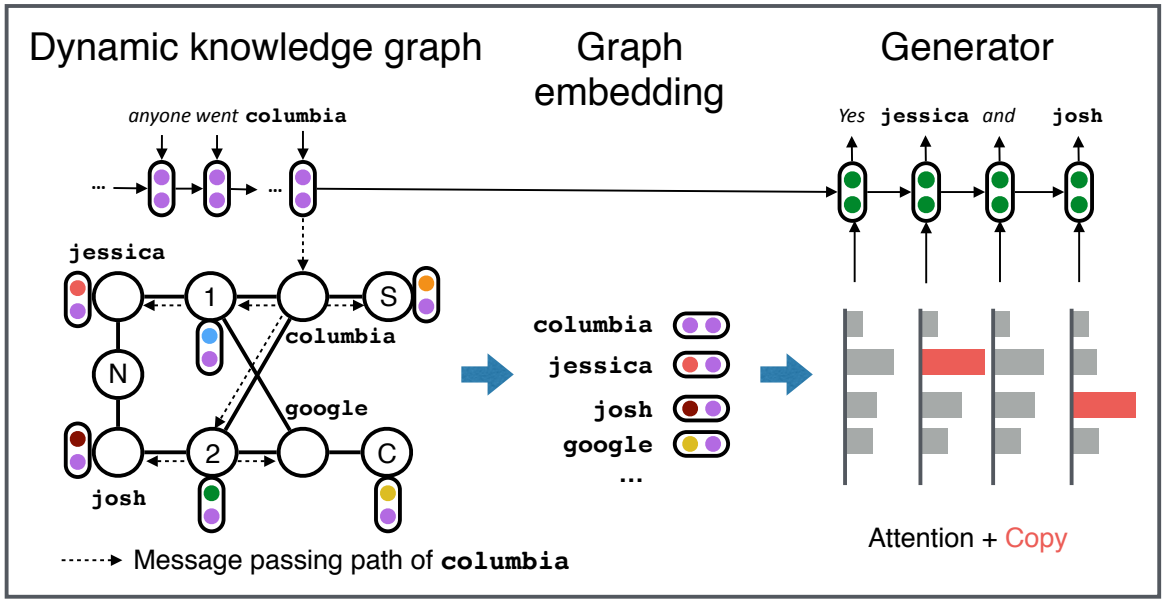
Figure 29.11 An LSTM architecture for slot filling, mapping the words in the input (represented as 1-hot vectors or as embeddings) to a series of IOB tags plus a final state consisting of a domain concatenated with an intent.

Frame-based & KB-based Dialog Agents

KB + Dialogue history

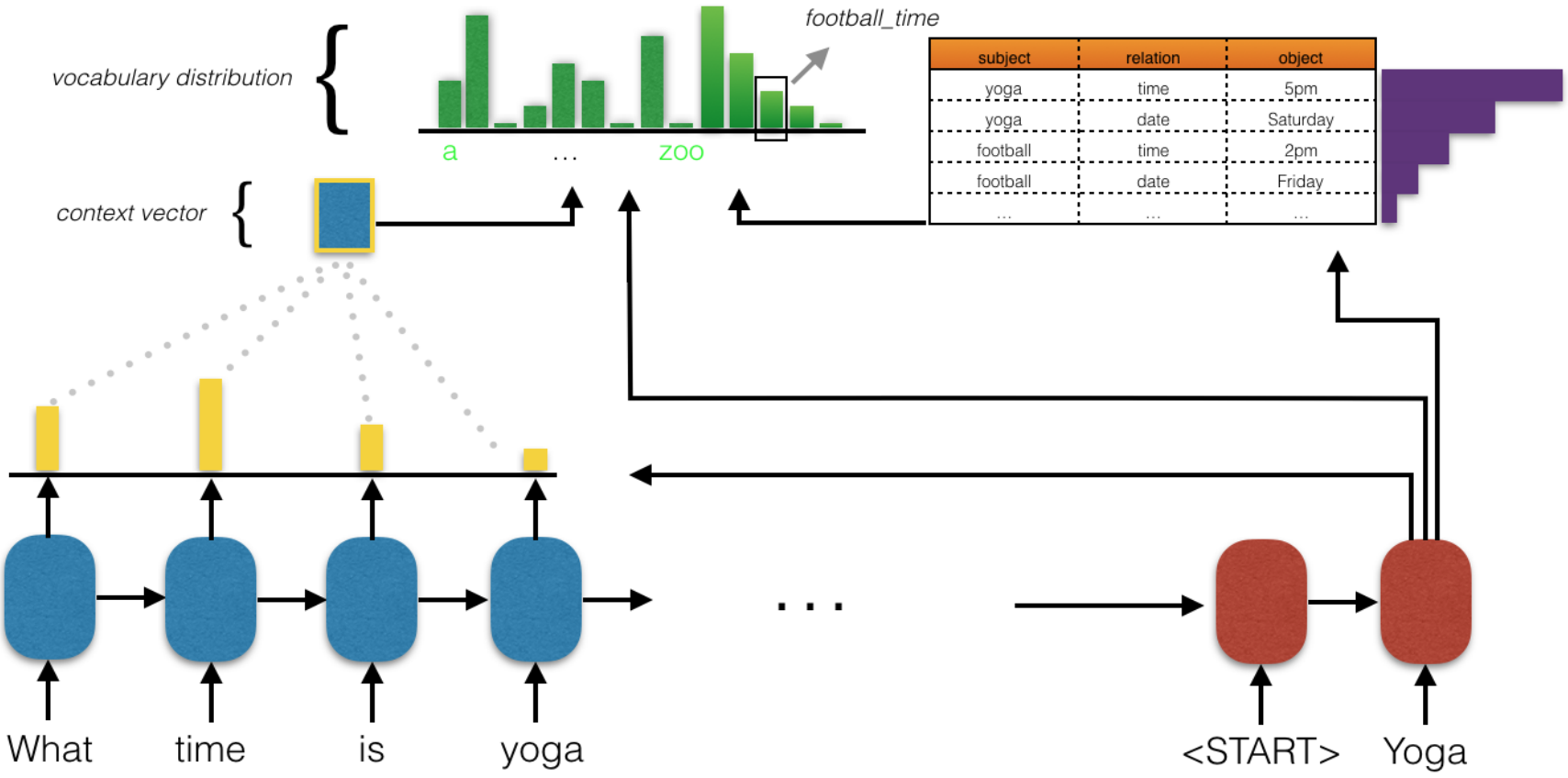
	Name	School	Company
Item 1	Jessica	Columbia	Google
Item 2	Josh	Columbia	Google

B: *anyone went to* **columbia**?



[He et al., 2017]

Frame-based & KB-based Dialog Agents



[Eric et al., 2017]