Language Generation

Announcements

- Reading: language generation paper for today
  - This is the baseline model for the E2E language generation challenge
  - Your HW4 implementation is based on this model
- Wed, Nov 20th: Or Biran, Elementary Cognition: Dialog systems.
- Final exam: In-class, Dec. 9th: see syllabus
- Monday, Nov 25th: Bias

Relevant news article

- We teach A.I. Systems Everything, Including our Biases

Beam search complexity

- Time complexity: linear because it only expands b nodes at each level
  - Worst case: O(Bm) where B is beam and m is maximum depth of any path
- Space complexity: linear
  - Worst case: O(Bm)
- Not optimal

Today

Extractive summarization of news articles

Language generation

- The E2E task
- Baseline model used in the task

Another neural summarization approach

- Extractive summarization of news
  - Single document summarization

- Data source: Daily News
  - Bulleted highlights of each article

- Neural Summarization by Extracting Sentences and Words
  - Cheng and Lapata, Edinburgh

Example from Daily News

AFL star blames vomiting cat for speeding

Adelaide Crows defender Daniel Talia has kept his driving license, telling a court he was speeding 36km over the limit because he was distracted by his sick cat.

The 22-year-old AFL star, who drove 96km/h in a 60km/h road works zone on the South Eastern expressway in February, said he didn't see the reduced speed sign because he was so distracted by his cat vomiting violently in the back seat of his car.

In the Adelaide magistrates court on Wednesday, Magistrate Bob Harrap fined Talia $824 for exceeding the speed limit by more than 30km/h.

He lost four demerit points, instead of seven, because of his significant training commitments.

- *Adelaide Crows defender Daniel Talia admits to speeding but says he didn't see road signs because his cat was vomiting in his car.*
- *22-year-old Talia was fined $824 and four demerit points, instead of seven, because of his 'significant' training commitments.*

Figure 1: DailyMail news article with highlights. Underlined sentences bear label 1, and 0 otherwise.

Cheng and Lapata 2016

Example from Daily News

AFL star blames vomiting cat for speeding

Adelaide Crows defender Daniel Talia has kept his driving license, telling a court he was speeding 36km over the limit because he was distracted by his sick cat.

The 22-year-old AFL star, who drove 96km/h in a 60km/h road works zone on the South Eastern expressway in February, said he didn't see the reduced speed sign because he was so distracted by his cat vomiting violently in the back seat of his car.

In the Adelaide magistrates court on Wednesday, Magistrate Bob Harrap fined Talia $824 for exceeding the speed limit by more than 30km/h.

He lost four demerit points, instead of seven, because of his significant training commitments.

- *Adelaide Crows defender Daniel Talia admits to speeding but says he didn't see road signs because his cat was vomiting in his car.*
- *22-year-old Talia was fined $824 and four demerit points, instead of seven, because of his 'significant' training commitments.*

Figure 1: DailyMail news article with highlights. Underlined sentences bear label 1, and 0 otherwise.

Paraphrasing
Compression
Fusion

Cheng and Lapata 2016

Two Tasks

- Input: Document D: $\{s_1,\ldots s_m\}$ consisting of words $w_1,\ldots.w_n$
- Sentence extraction
  - Select a subset of j sentences, j<m
  - Score each sentence and predict label y_L ε {0,1}
  - Objective: Maximize all sentence labels given D and weights θ
- Word extraction
  - Find a subset of words in D and their optimal ordering
  - Language generation task with output vocabulary restricted to input D vocabulary
  - Objective: Maximize the likelihood of generated sentences, further decomposed by considering conditional dependencies among their words

Training Data

- Sentence extraction
  - Highlights are abstracts
  - Find the s in D that most closely matches a highlight sentence
    - Positive, unigram and bigram matches, #entities
  - 200K document/summary pairs, summary size = 30% document
- Word extraction
  - Retain highlights with all words from D
  - Find neighbors of words not in D and substitute
  - 170K document/summary pairss

Neural Summarization Architecture

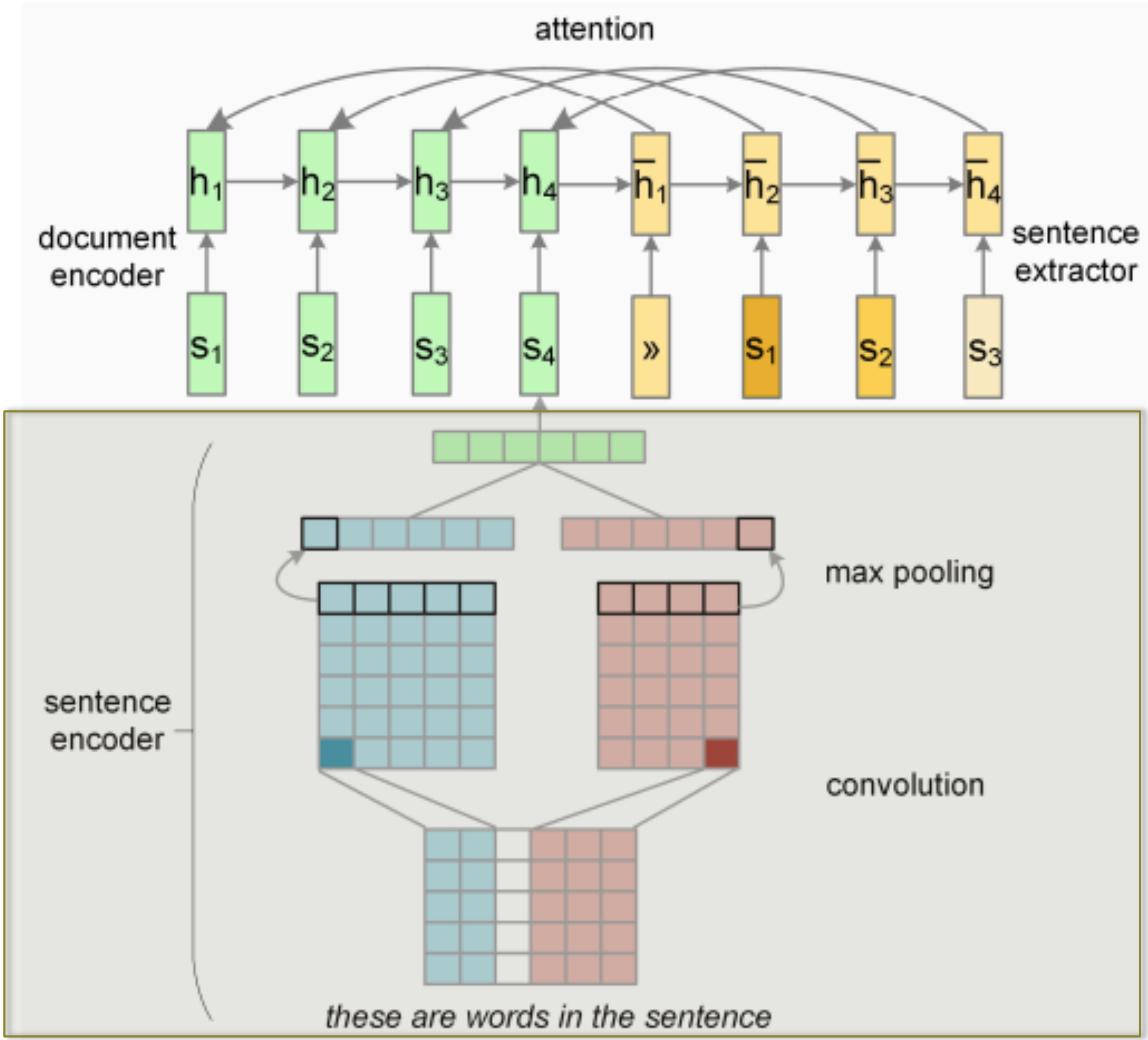
- Hierarchical document reader
  - Derive meaning representation of document from its constituent sentences

- Attention based hierarchical content extractor

- Encoder-decoder architecture

Document Reader

- CNN sentence encoder
  - Useful for sentence classification
  - Easy to train
- LSTM document encoder
  - Avoids vanishing gradients



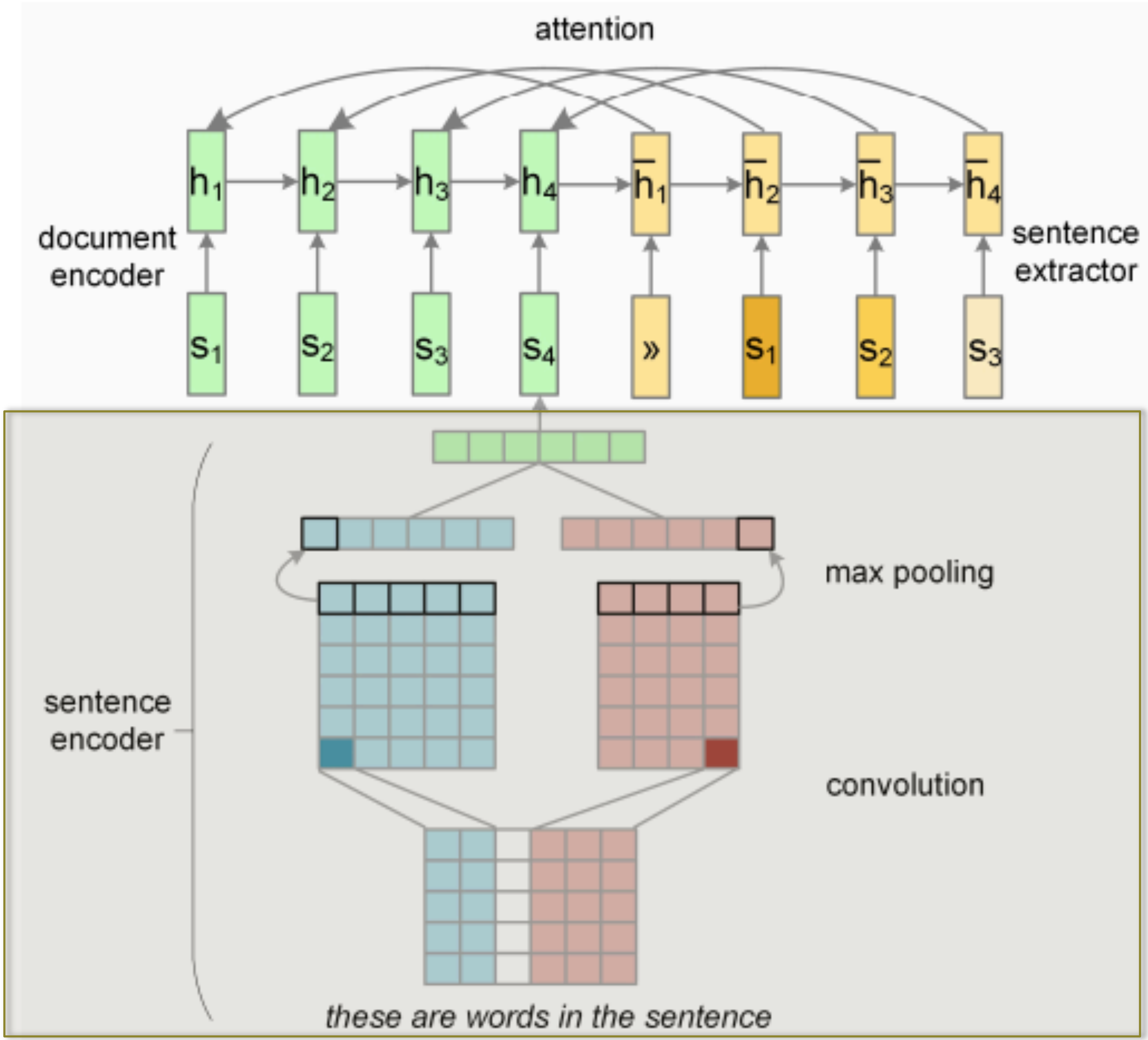
Cheng and Lapata 2016

CNN

$$\mathbf{f}^i_j = \tanh(\mathbf{W}_{j:j+c-1} \otimes \mathbf{K} + b)$$

- Where W $\varepsilon \mathbb{R}^{nXd}$ and d = word embedding dimension, n = #words in sentence
- K a kernel of width c, b the bias
- f^i_j = the jth item in the ith feature map f^i
- Perform max pooling over time to obtain a single feature to represent the sentence

$$\mathbf{s}_{i,\mathbf{K}} = \max_j \mathbf{f}^i_j$$



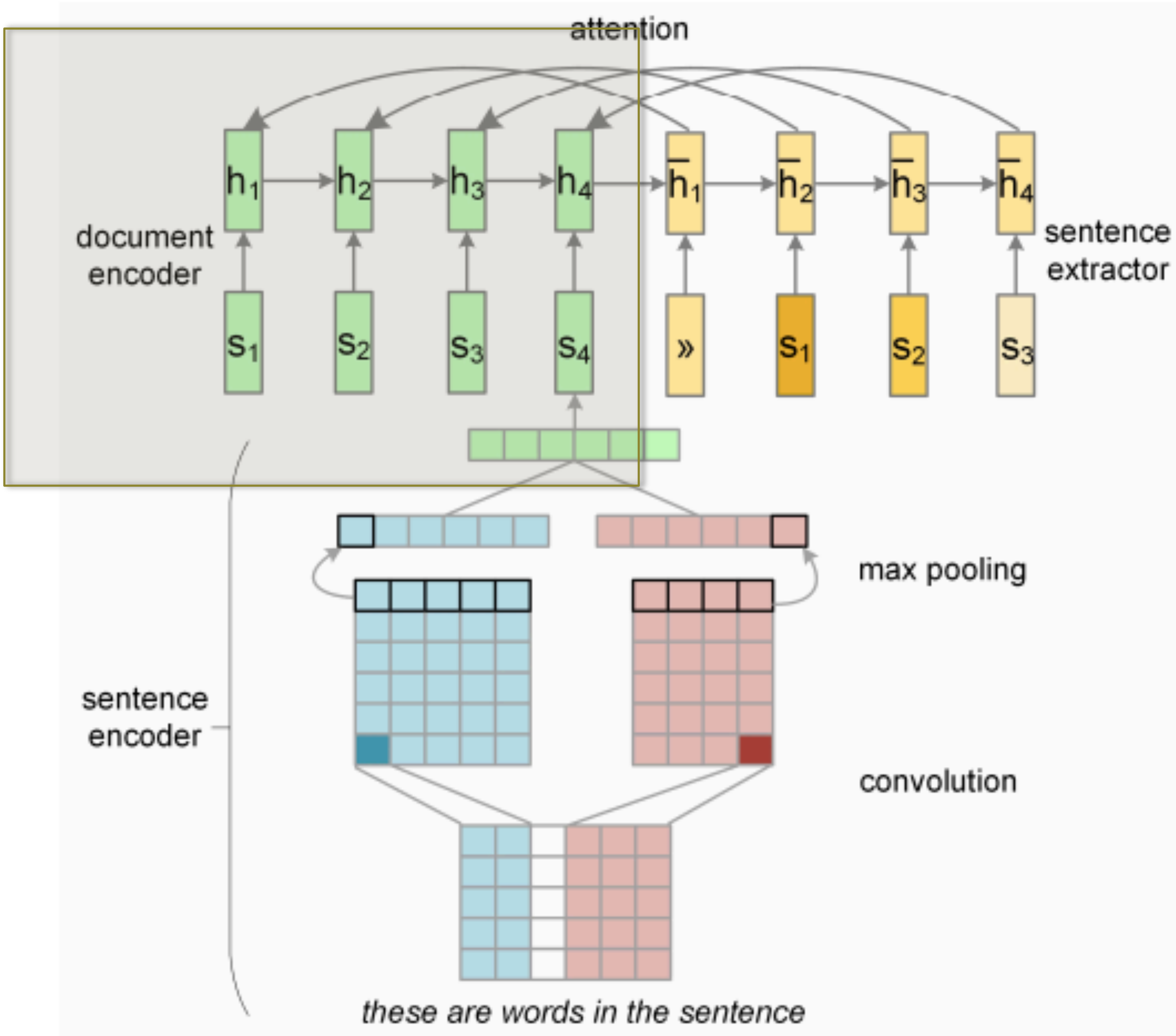
Cheng and Lapata 2016

Recurrent document encoder

- LSTM to compose a sequence of sentence vectors into a document vector

- The hidden states of the LSTM = a list of partial representations
  - Each focuses on the corresponding input sentence given previous content

- Altogether constitute document representation



Cheng and Lapata 2016

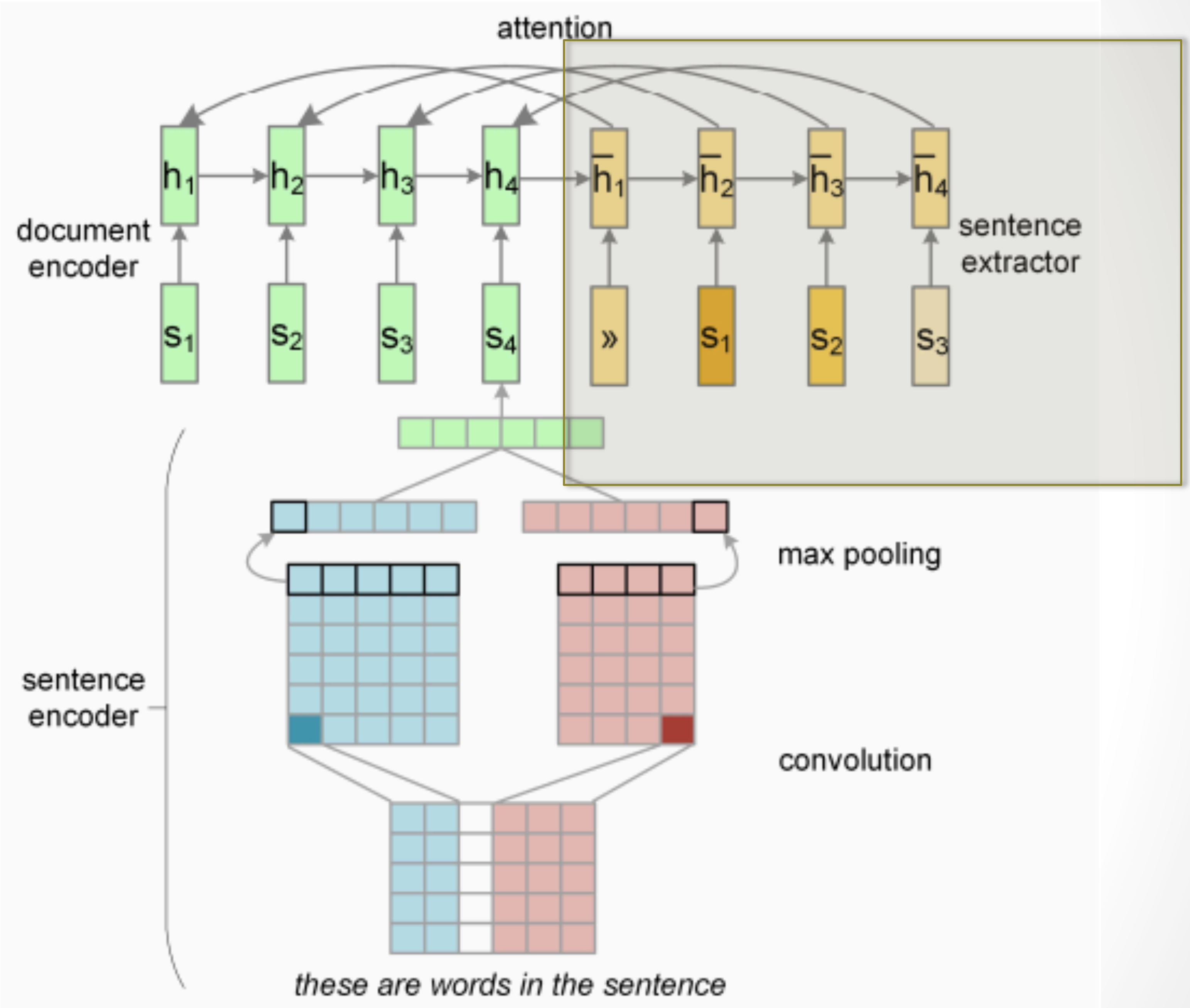
Sentence Extractor

- Applies attention to directly extract sentences after reading them

$$\bar{\mathbf{h}}_t = \mathrm{LSTM}(p_{t-1}\mathbf{s}_{t-1}, \bar{\mathbf{h}}_{t-1})$$

$$p(y_L(t) = 1|D) = \sigma(\mathrm{MLP}(\bar{\mathbf{h}}_t : \mathbf{h}_t))$$

- ħ extractor hidden state, h encoder hidden state
  - Attends to relation between extractor and encoder hidden state
- MLP takes as input concatenated ħ and h
- P_{t-1} degree to which extractor believes previous sentence should be extracted



Cheng and Lapata 2016

Word Extractor

- Instead of extracting sentence, extracts next word
- Uses hierarchical attention to attend to sentence and word within sentence
- Output vocabulary restricted to input sentence

- -> conditional language model with vocabulary constraint

Datasets

- Daily Mail
  - 200K training
  - 500 test
- DUC 2002
  - 567 documents with 2 summaries each

Results

DUC 2002	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	43.6	21.0	40.2
LREG	43.8	20.7	40.3
ILP	45.4	21.3	42.8
NN-ABS	15.8	5.2	13.8
TGRAPH	48.1	**24.3**	—
URANK	**48.5**	21.5	—
NN-SE	47.4	23.0	**43.5**
NN-WE	27.0	7.9	22.8

DailyMail	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	20.4	7.7	11.4
LREG	18.5	6.9	10.2
NN-ABS	7.8	1.7	7.1
NN-SE	**21.2**	**8.3**	**12.0**
NN-WE	15.7	6.4	9.8

Table 1: ROUGE evaluation (%) on the DUC-2002 and 500 DailyMail samples.

Later results

(Nallapati et al 2017):

- RNN over sentence embeddings and output concatenated, representation of entire document by averaging, representation of summary so far by summing outputs

- (Kedzie, McKeown and Daume 2018): simpler is better
  - Averaging word embeddings, pre-trained fine, no need for summary so far
  - Order most important for news

State of the Art

- http://nlpprogress.com/english/summarization.html
- Is this a good task?
- Could you imagine other summarization tasks for which there might be data?

Language Generation

- The E2E Challenge
  - 62 submissions by 17 institutions, 11 countries, 1/3 from industry
- Restaurant recommendations
- Generation of one or more sentences from an input meaning representation (MR)
- Large and varied dataset

Input Data

- Unordered sets of attributes
- MR:
  - Name[The Wrestlers], pricerange[cheap], customerrating[1 of 5]
- Output
  - The Wrestlers offer competitive prices but it isn't highly rated by customers.

Domain Ontology

Attribute	**Data Type**	**Example value**
name	verbatim string	*The Eagle, ...*
eatType	dictionary	*restaurant, pub, ...*
familyFriendly	boolean	*Yes / No*
priceRange	dictionary	*cheap, expensive, ...*
food	dictionary	*French, Italian, ...*
near	verbatim string	*market square, Cafe Adriatic, ...*
area	dictionary	*riverside, city center, ...*
customerRating	enumerable	*1 of 5 (low), 4 of 5 (high), ...*

Dusek et al 2018
https://arxiv.org/pdf/1901.07931.pdf

How was data gathered

- Crowd sourcing on CrowdFlower
- Experiment with 2 kinds of prompts
  - List of randomly ordered attributes
  - Pictorial representations
- Payment: .02/page containing 1 MR, 20 seconds/hit
  - See https://www.nytimes.com/interactive/2019/11/15/nyregion/amazon-mechanical-turk.html

name[Loch Fynne] eatType[restaurant]
familyFriendly[yes]
priceRange[cheap]
foodType[Japanese]

Picture: Serving low-cost Japanese style cuisine, Loch Fynne caters for everyone, including families with small children.

name[The Wrestlers]
familyFriendly[no]
area[The River]
Food[Italian]
customerRating[5 of 5]
priceRange[expensive]
Near[Café Adriatic]
eatType[restaurant]

Dusek et al 2018
https://arxiv.org/pdf/1901.07931.pdf

Training Examples

- Crowd sourced ~50K instances
  - 6K MRs
  - 5 slots/MR
  - Largest dataset of its kind
    - Sfrest: 5K instances, 1K MR
    - Bagel: 404 instances, 380 MR
- Average of 8.27 references per MR

Delexicalization

- **MR:** name[Green Man], food[French], priceRange[more than 30 pounds], area[city centre], familyFriendly[no], near[All Bar One]
- **Lex:** ***Green Man*** is a French restaurant in the city centre. It is not child friendly and is located near ***All Bar One***. It costs more than thirty pounds.
- **Delex:** ***X-name*** is a french retaurant in the city centre. It is not child friendly and is located near ***X-near***. It costs more than thirty pounds.

Traditional language generation

- Content selection
  - (Done for us in the E2E task)

- Aggregation
  - Which pieces of content go into which sentence?

- Realization
  - How does a tree get realized in English?

The baseline system

- Two step generation
  - Sentence planning and surface realization are separated
- Joint one-step approach
  - Directly produces a natural language string

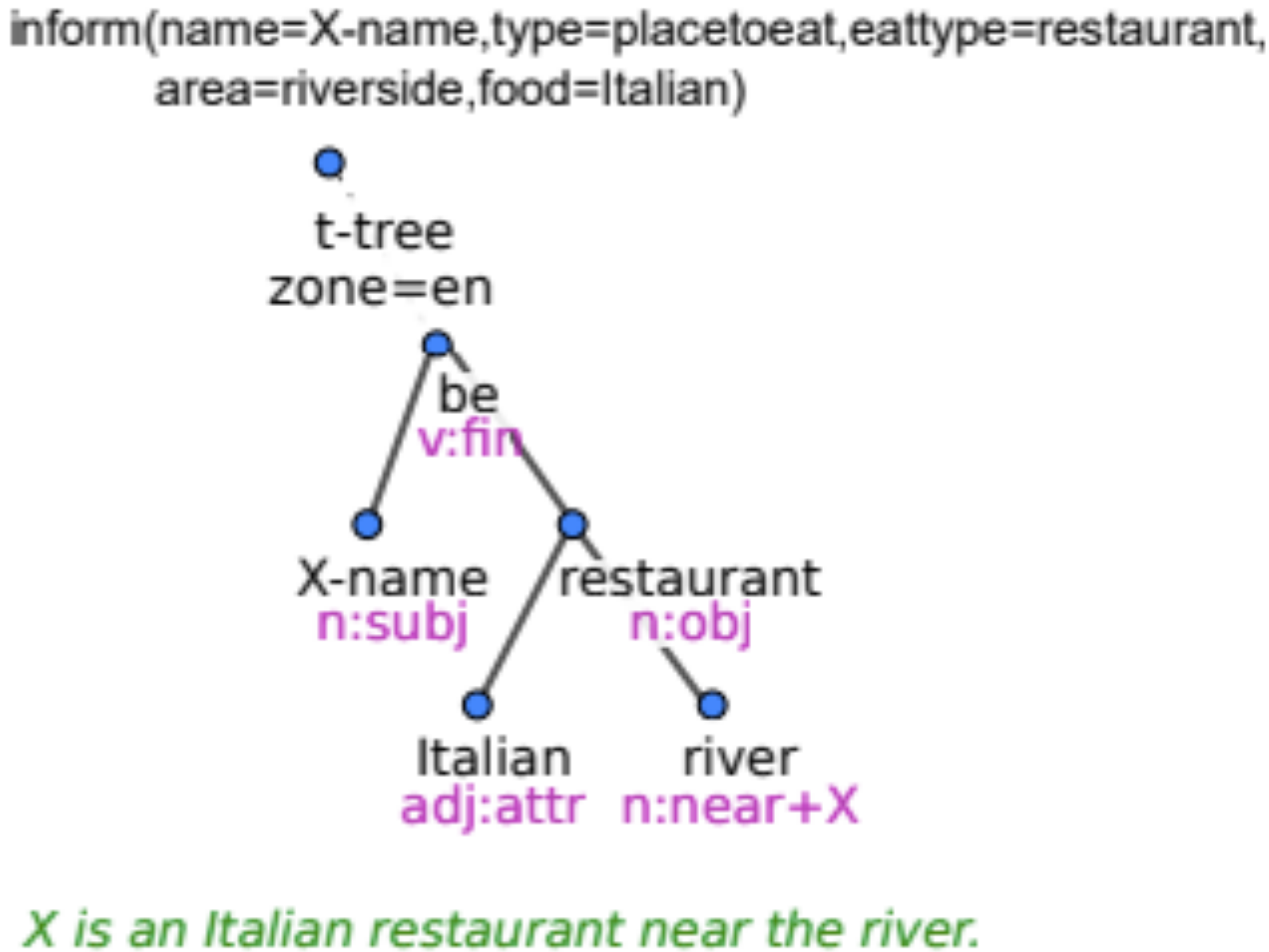


Figure 1: Example DA (top) with the corresponding deep syntax tree (middle) and natural language string (bottom)

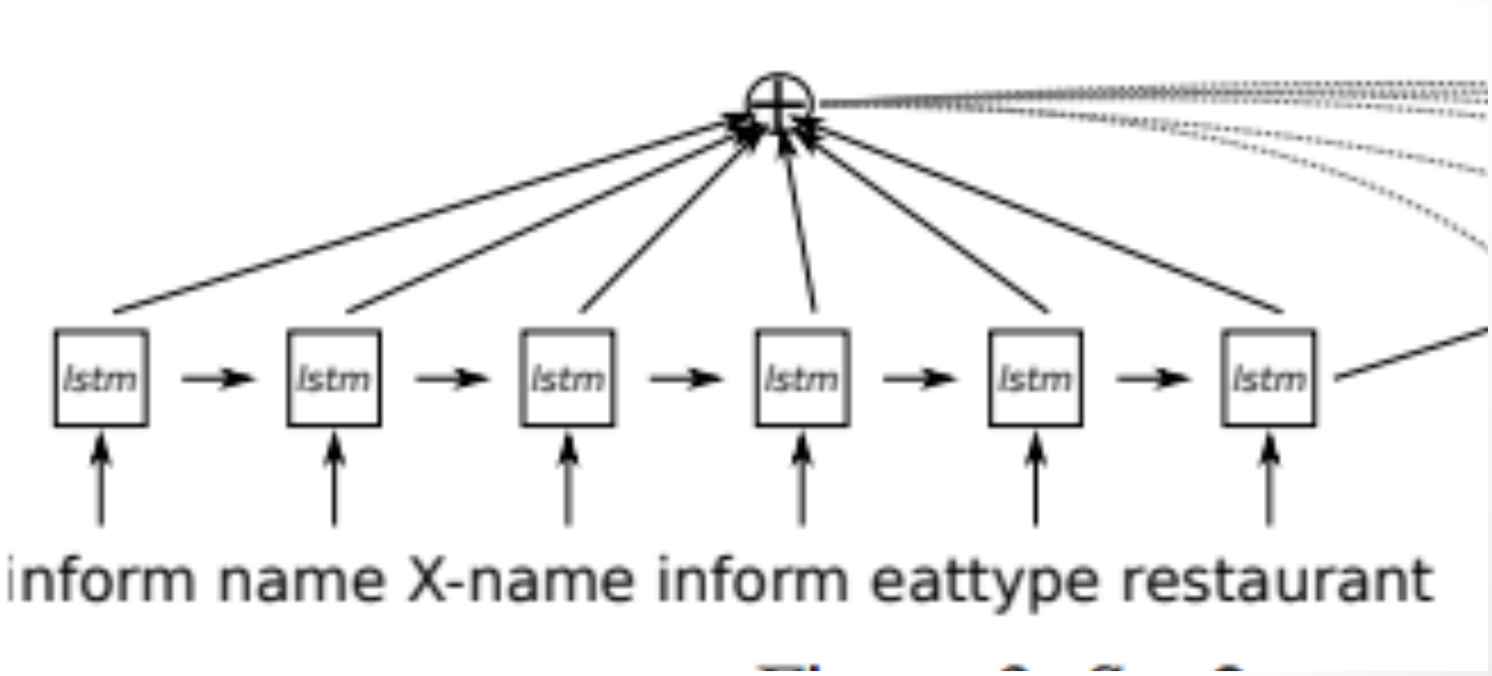
Seq2seq model

- Encoder decoder RNN (Cho et al 2014, Sutskever et al 2014)

- Need to convert input dialog act (DA) and output tree into sequences

DA: Sequence representation

- Triple: DA type, slot, value
- Concatenate triples all slots
- Each token is an embedding



Syntax trees as sequences

- (<root> <root> ((X-name n:subj) be v:fin ((Italian adj:attr) restaurant n:obj (river n:near+X)))

Seq2seq model

Figure 3: Seq2seq generator with attention

- Encoder
  - $X = \{x_1, x_2, \ldots x_n\}$
  - RNN to encode into a sequence of encoder output/hidden states $h = \{h_1, h_2, \ldots. h_n\}$
    - Where $h_t = \text{lstm}(x_t, h_{t-1})$

Dusek and Jurcicek 2016
https://www.aclweb.org/anthology/P16-2008.pdf

Seq2seq model

Figure 3: Seq2seq generator with attention

- Decoder
  - Output $y = \{y_1, y_2, \ldots y_n\}$
  - $P(y_t | y_1 \ldots . y_{t-1}, x) = \mathrm{softmax}((s_t \circ c_t)\ W_y)$
  - S_t is the decoder state
    - $S_0 = h_n$
    - $S_t = \mathrm{lstm}(((y_{t-1} \circ c_t)W_S, s_{t-1})$

Dusek and Jurcicek 2016
https://www.aclweb.org/anthology/P16-2008.pdf

Beam search in this context

- Last time:

  Filter k-max

  - $\pi\ (i+1) <- \text{K-argmax}\ g(y_{i+1},y_c,x) + s(y,x)$

  - What was y_c?
  - What would we use here in place of y_C?

Beam search and re-ranker

- Most common errors (semantic errors)
  - Missing an attribute
  - Added an attribute (hallucination)
  - Wrong value for an attribute
- Re-ranker scores n-best output by penalizing those that added or missed an attribute
  - Vector of realized attributes compared to vector of input attributes
  - Hamming distance is the penalty
- Learn a classifier for each output over the scores
  - Sigmoid $(h_n \bullet \quad W_R + b)$

Computing Hamming distance

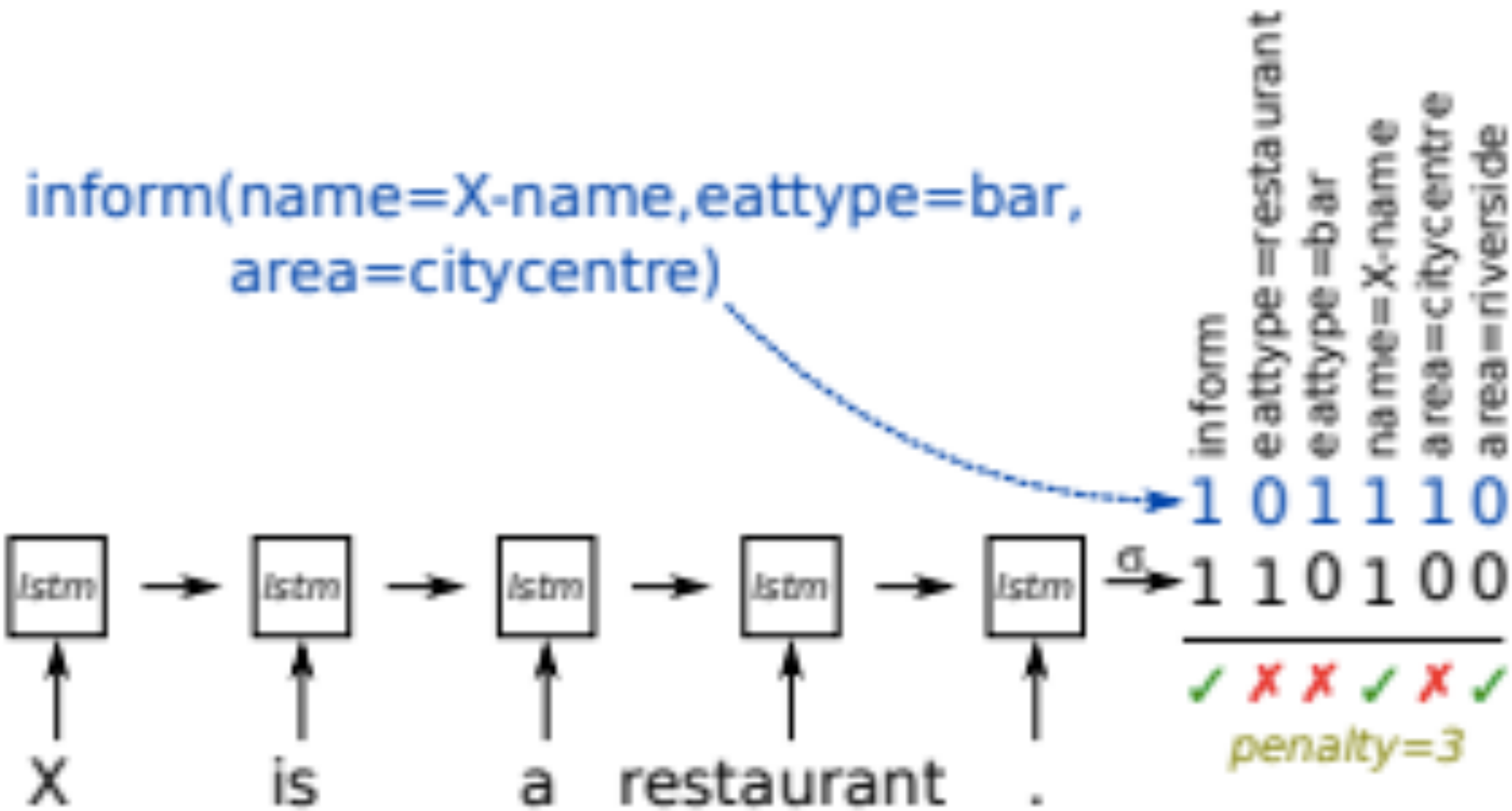


Figure 4: The reranker

Beam search on your homework

- Suggest using log likelihood normalized by length

- If you'd like to do something more sophisticated, such as this, can earn you extra credit

Setup	BLEU	NIST	ERR
Mairesse et al. (2010)*	~67	-	0
Dušek and Jurčíček (2015)	59.89	5.231	30
Greedy with trees	55.29	5.144	20
+ Beam search (b. size 100)	58.59	5.293	28
+ Reranker (beam size 5)	60.77	5.487	24
(beam size 10)	60.93	5.510	25
(beam size 100)	60.44	5.514	19
Greedy into strings	52.54	5.052	37
+ Beam search (b. size 100)	55.84	5.228	32
+ Reranker (beam size 5)	61.18	5.507	27
(beam size 10)	62.40	5.614	21
(beam size 100)	62.76	5.669	19

Table 1: Results on the BAGEL data set

Dusek and Jurcicek 2016

E2E 2018 Challenge Take-aways

- Seq2seq score high on automatic metrics and human evaluations of naturalness
  - Other approaches: statistical/ML, template filling (learned and manual)
- But seq2seq often fail to correctly express a meaning representation
- Seq2seq can be outperformed by hand-engineered
  - On overall quality, complexity, length and diversity of output
- https://arxiv.org/pdf/1901.07931.pdf