Final Class

Announcements

- Take a few minutes for the evaluation at the beginning of class
- Grades: expect to post final grades by 12/23
- Several funded GRA positions open to MS students next spring – if interested email me.
- HW3 grades returned today
- Final exam will be in two locations. You will receive an email about where.
- **DO NOT ANSWER POLL EVERWHERE AHEAD OF TIME**

HW4

- Instructions for submitting: The directory structure on the bitbucket repo should be exactly same as the hw4.zip provided to you (with the exception of data directory. Do not upload it). To push the code to remote repo, use the same instructions as given in HW0. Double check you remote repo for correct directory structure. ***We won't consider any regrade requests based on wrong directory structure penalty. Again, do not upload data to your bitbucket repo***.

- HW4: Use log of weights in attention to get slightly better visualization. See post by Dheeraj.
- Apoorv has office hours **TODAY 4-6** for those with questions.
- Written part: Not just talking about small extensions to neural net. Think big.

Projects in NLP

- Search and summarization over low resource languages
  - How do we summarize in the source? How do we summarize when translation is bad? How do we summarize speech?
- Identifying aggression and loss in posts from gang-involved youth
  - Can we identify patterns in posts over time? Can we use the social network? Can we identify references to triggering events?
- Identifying hate speech
  - Bullying, threats against journalists, how does culture affect interpretation?
- Joint use of visual and textual cues
  - To identify sentiment towards targets, to identify events

Reflections

Today

- Poetry Generation
- Review for final

Poetry Generation

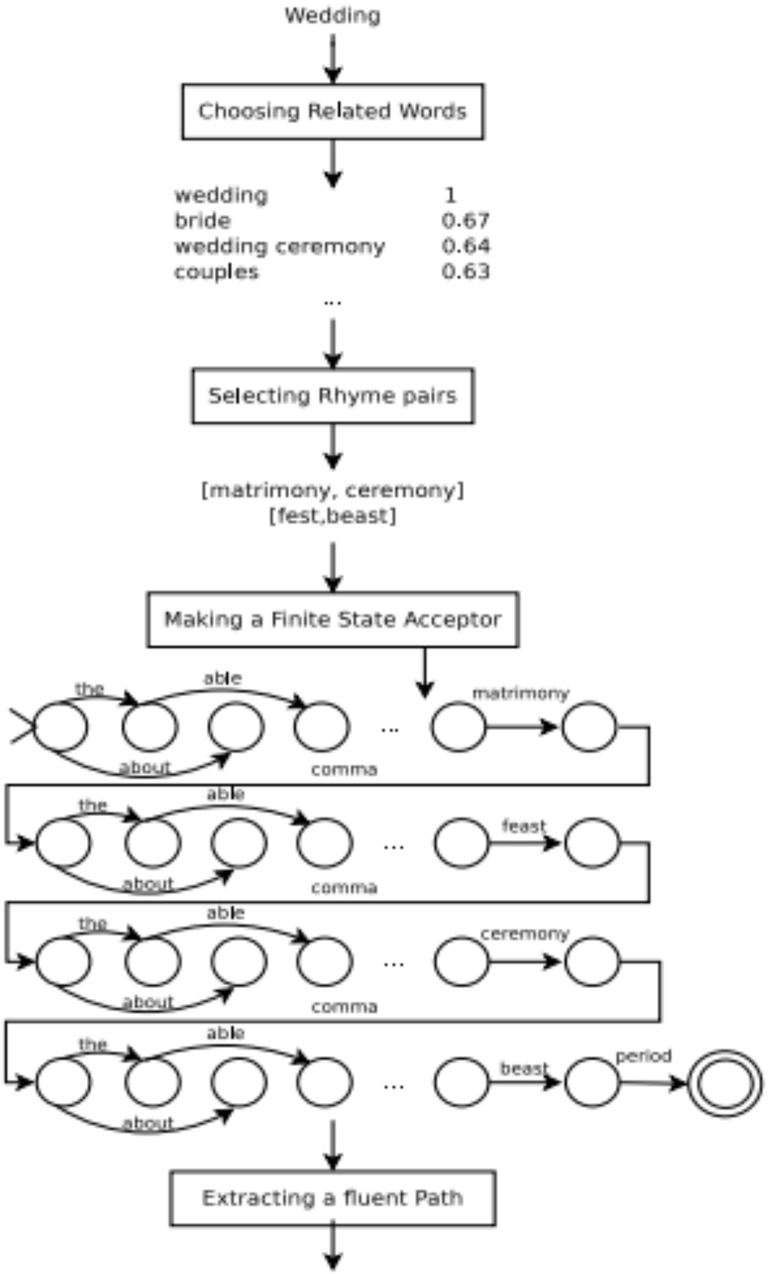
- Generating Topical Poetry, Ghazvininejad et al 2016) https://aclweb.org/anthology/D16-1126

- Hafez – generates sonnets on a user provided topic
  - Iambic pentameter
  - Every other line rhymes

- Rough overview of methods plus output

System Overview

- Select large vocabulary and compute *stress* patterns
- Select *words related to user-supplied topic*
- Select *pairs of rhyming words* to end lines
- Build *FSA* with a path for every conceivable sequence of vocabulary words to obey *formal rhythm constraints with rhyme words in place*
- Select a *fluent path* through the FSA using a *RNN for scoring*



The greatest gift of holy matrimony,
Declare an order from a wedding feast,
Or open up a wedding ceremony,
And hail the son of God and kill the beast.

Ghazvininejad et al 2016)

Vocabulary

- Iambic pentameter: ten syllables alternating between stressed and unstressed
  - *Attending on his golden pilgramage*
    010 1 0 10 101
- Use CMU pronunciation dictionary
  - Remove words whose stress pattern does not match iambic pattern
  - Remove ambiguous words (*record N 10*; *record V 01*)
  - Avoids *to, it, in, is*
- Final vocabulary: 14368 words
  - 4833 monosyllabic
  - 9535 multisyllabic

Selecting topically related words

- User supplies a topic: *colonel*
- Output: *colonel, lieutenant_colonel, brigadier_general, commander, army*

- Use Word2Vec using window size of 40
  - Word embedding vector for topic word or phrase
  - Word embeddings for each vocabulary word

- How would similarity be computed?

Rhyme words

- Shakespearean sonnet: ABAB CDCD EFEF GG
- Strict rhyme: sounds of two words must match from the last stressed vowel onwards
  - Masculine rhyme: the last syllable is stressed
  - Feminine rhyme: the penultimate syllable is stressed
  - Pre-compute strict rhyme classes for words and hash vocabulary into those classes (CMU pronunciation dictionary)
- Slant rhymes: *viking/fighting, snoopy/spooky, baby/crazy and comic/ironic*

Rhyme word selection

- Hash all related words/phrases into rhyme classes
- Each collision generates a candidate rhyme pair (s1, s2)
- Score pair with max: cosine (s1, topic); cosine (s2, topi)
- Choose rhyme pairs randomly with probability proportional to their score

FSA Construction

- Create large FSA that encodes all word sequences that use selected rhyme pairs and obeys formal sonnet constraints
  - Contains 14 lines
  - Lines are in iambic pentameter with stress pattern $(01)^5$ or $(01)^5 0$ (feminine)
  - Each line ends with chosen rhyme word/phrase
  - Each line is punctuated with comma or period except for 4th, 8th, 12th and 14th which have a period

FSA Output

- Topic: natural language
- Contains 10^{229} paths
- Randomly selected path
  - *Of pocket solace ammunition grammar.*
  - *An tile pretenders spreading logical.*
  - *An stories Jackie gallon posing banner.*
  - *An corpses Kato biological…*

Path extraction through FSA with RNN

- Need scoring function and search procedure
- RNN “generation model”
  - Two layer LSTM with beam search guided by FSA
    - Beam search state: (h, s, word, score)
      - H the hidden state of LSTM at step t in the ith state
      - S the FSA state at step t in the ith state
      - Generates one word at each step
- Trained using song lyrics -> repeating words (never ever ever ever ever)
  - Apply penalty to words already generated
- Beam of 50 often results in not being able to generate final rhyming word in FSA
  - Generate the whole sonnet in reverse

Translation model

- Use encoder-decoder LSTM
- Assemble rhyming words in reverse order (encoder side)
- Paired with entire reversed lyric (decoder side)
- At generation time: all selected rhyme words on source side and let model generate the poem conditioned on those rhyme words
- When generating the last line, it already knows all 14 rhyme words

Results

- Translation model better than generation model
- Encouraging topic words better than not
- Does the system plagiarize?
  - Average copying of 1.2 5-grams per sonnet
  - If relax repeated-word penalty -> 7.9 copied 5-grams
  - If relax iambic meter -> 10.6 copied 5-grams

Bipolar Disorder

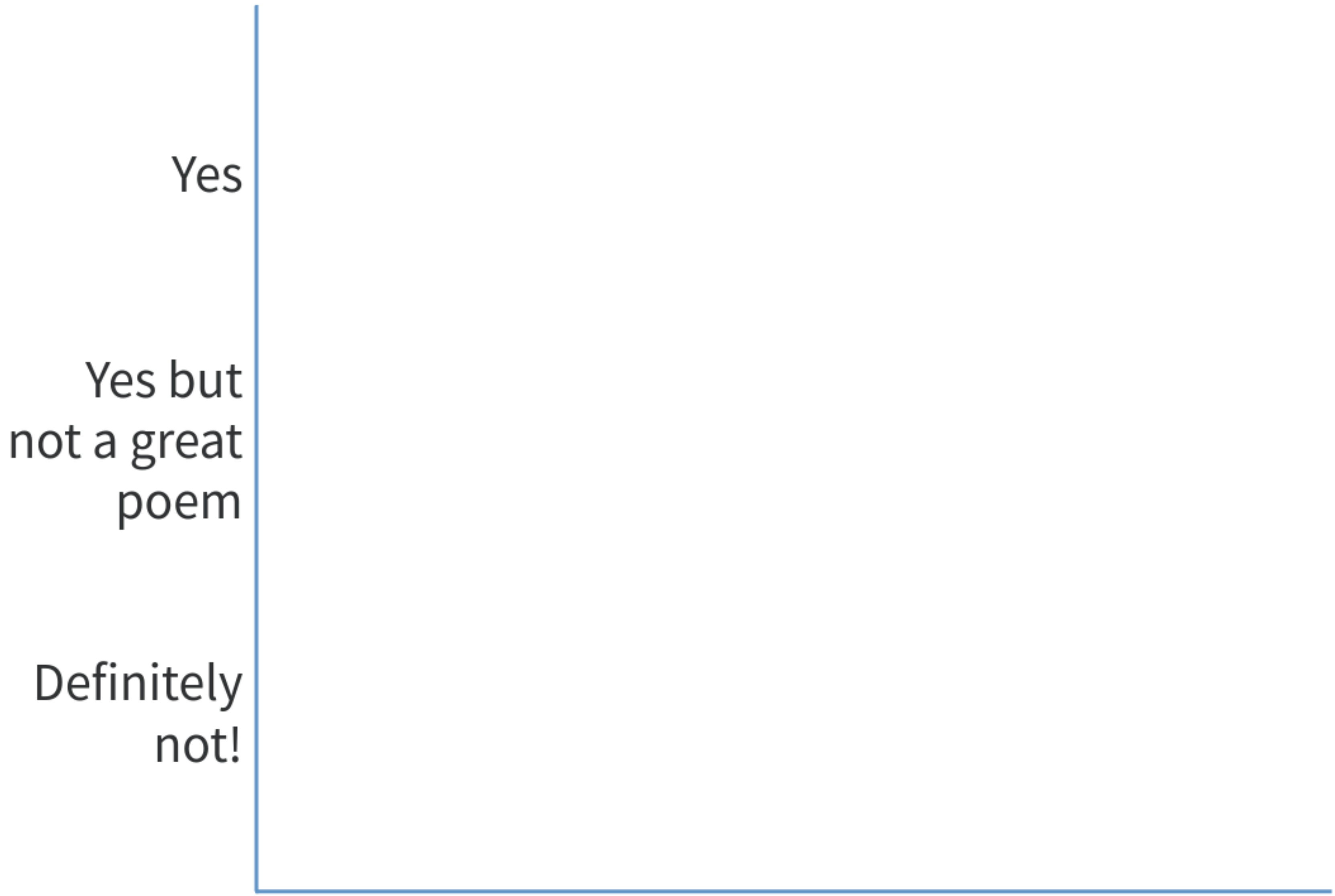
- *Existence enters your entire nation.*
  A twisted mind reveals becoming manic,
  An endless modern ending medication,
  Another rotten soul becomes dynamic.

- *Or under pressure on genetic tests.*
  Surrounded by controlling my depression,
  And only human torture never rests,
  Or maybe you expect an easy lesson.

- *Or something from the cancer heart disease,*
  And I consider you a friend of mine.
  Without a little sign of judgement please,
  Deliver me across the borderline.

- *An altered state of manic episodes,*
  A journey through the long and winding roads.

This passes the Turing test for a poem



Other topics

- *Love at First Sight*

  An early morning on a rainy night,
  Relax and make the other people happy,
  Or maybe get a little out of sight,
  And wander down the streets of Cincinnati.

Girlfriend

- *Another party started getting heavy.
  And never had a little bit of Bobby,
  Or something going by the name of Eddie,
  And got a finger on the trigger sloppy.*

Noodles

- *The people wanna drink spagetti alla,
  And maybe eat a lot of the other crackers,
  Or sit around and talk about the salsa,
  A little bit of nothing really matters.*

Final Review

- Final exam will be in two locations. You will receive an email about where
- Today: after midterm only. Look at midterm review to review earlier topics
- Final will be cumulative
  - Some emphasis towards last half of the class
  - Some emphasis towards topics not tested in homeworks
  - Anything covered in class is a potential topic for the exam
- Calculator allowed in final. No other electronics. No notes or books.
- Three review sessions will be offered by Tas
  - Bhavana, December 13th, Neural Net basics, HW3
  - Elsbeth, December 18th, evening
  - Fei-Tzin, December 19th, evening
- All office hours will be held between now and final

Today

- Semantics
- RNN, LSTM, Attention
- Summarization
- Machine Translation
- Questions

Abstract Meaning Representation

- Given a sentence propose a representation
- Given a representation, provide the sentence
- Understand the parsing framework
- Discuss pros and cons

AMR characteristics

- Rooted, labeled graphs
- Abstract away from syntactic differences
  - He described her as a genius
  - His description of her: genius
  - She was a genius according to his description
- Use Propbank framesets
  - “bond investor”: invest-01
- Heavily biased towards English

AMR relations

- ~100 relations
- Frame arguments
  - Arg0, arg1, arg2, arg3, arg4, arg5 (Propbank)
- General semantic relations
  - :Accompanier, :age, :beneficiary, :cause, :compared-to, :concession, :condition, :consistof, :degree, :destination, :direction, :domain, :duration, :employed-by, :example, :extent, :frequency, :instrument, :li, :location, :manner, :medium, :mod, :mode, :name, :part, :path, :polarity, :poss, :purpose, :source, :subevent, :subset, :time, :topic, :value.
- Relations for quantity
  - :quant, :unit, :scale
- Relations for date entity
  - :day, :month, :year, :weekday, :time, :timezone, :quarter,:dayperiod, :season, :year2, :decade, :century, :calendar, :era.
- Relations for lists
  - :op1, :op2, …. :op10
- Plus inverses (e.g., :arg0-of, :location-of)

AMR relations

- ~100 relations
- Frame arguments

NOT NECESSARY TO MEMORIZE – WOULD BE PROVIDED

  by, :example, :extent, :frequency, :instrument, :li, :location, :manner, :medium, :mod, :mode, :name, :part, :path, :polarity, :poss, :purpose, :source, :subevent, :subset, :time, :topic, :value.
- Relations for quantity
  - :quant, :unit, :scale
- Relations for date entity
  - :day, :month, :year, :weekday, :time, :timezone, :quarter,:dayperiod, :season, :year2, :decade, :century, :calendar, :era.
- Relations for lists
  - :op1, :op2, …. :op10
- Plus inverses (e.g., :arg0-of, :location-of)

Framesets

- Examples of using Framesets to extract away from English syntax
- (d / describe-01
  - :arg0 (m/man)
  - :arg1 (m2 / mission)
  - :arg2 (d /disaster))
- :arg0 the describer, :arg1 the thing described, :arg2 what it is describing
- The man described the mission as a disaster. As the man described it, the mission was a disaster

Questions

- Amr-unknown to indicate wh-questions

- (f /find-01
  :arg0 (g /girl)
  :arg1 (a / amr-unknown))

  What did the girl find?

Compositionality

- The meaning of the whole is equal to the sum of the meaning of its parts
- How is AMR compositional?

  (d / describe-01
    - :arg0 (m/man)
    :arg1 (m2 / mission)
    :arg2 (d /disaster))

- (s / spy
      :arg0-of (a / attract-01))

- *What is the AMR for*

  the attractive spy described the mission as a disaster?

Learning to Search (L2S)

- Family of approaches that solves structured prediction problems
  - Decomposes the production of the structured output in terms of explicit search space
  - Learns hypotheses that control a policy that takes actions in the search space

- AMR is a structured semantic representation

- Model learning of concepts and relations in a unified setting.

AMR parsing task decomposed

- Predicting concepts
- Predicting the root
- Predicting relations between predicated concepts

Search space

- State s = $\{x_1,x_2,....,x_n,y_1,y_2,....,y_{i-1}\}$ where the input $\{x_1,x_2,....,x_n\}$ are the n words of the sentence

- Concept prediction: labels $y_1,y_2,....,y_{i-1}$ are the concepts predicted up to i-1.
  - Next action: y_i is the concept for word x_i from a k-best list of concepts

- Relation prediction: labels are relations for predicted pairs of concepts

- Root prediction: multi-task classifier selects root concept from all predicted concepts

Example

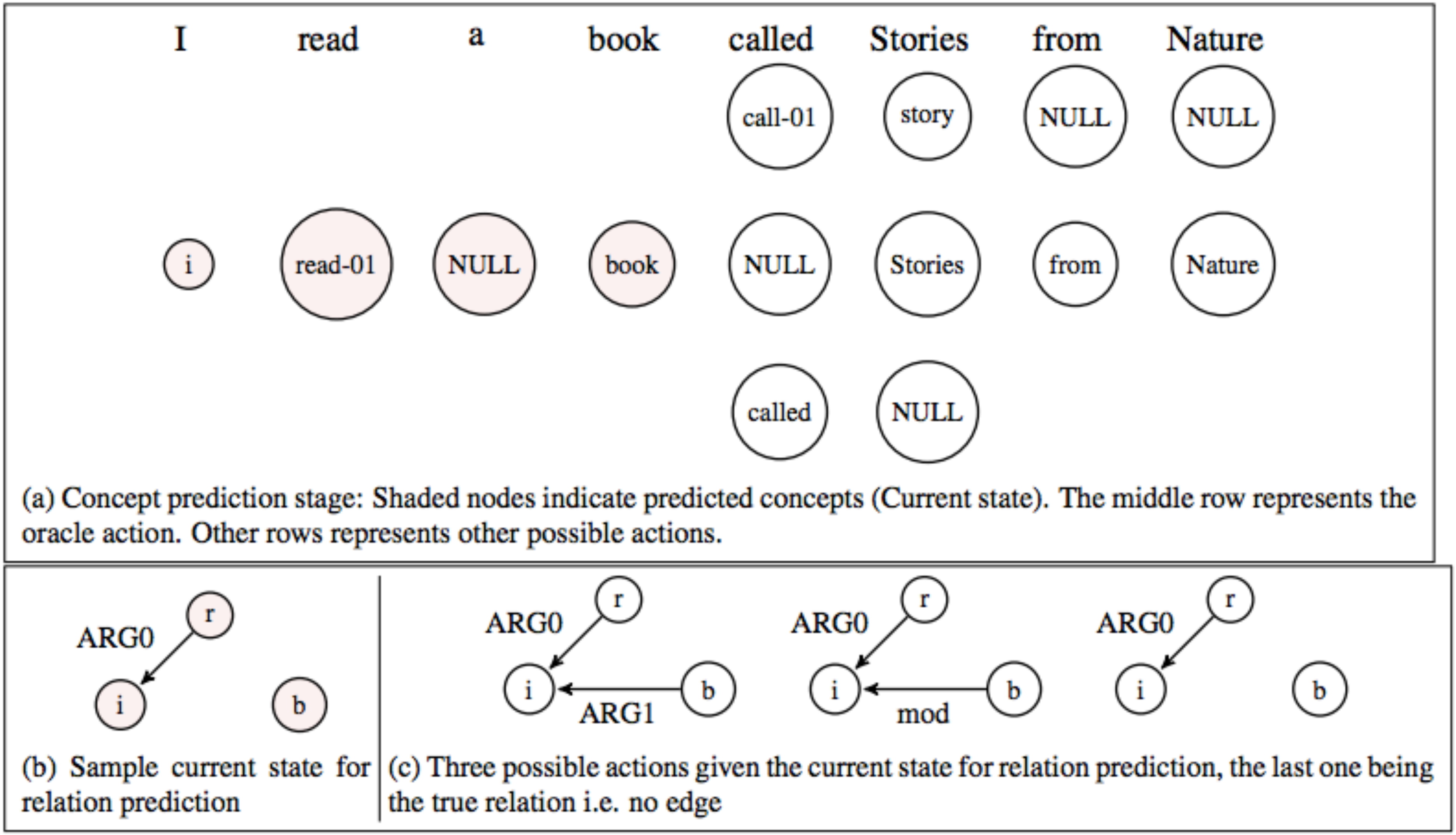


Figure 2: Using DAGGER for AMR parsing

Word Embeddings,
Distributional Semantics
Word Disambiguation
Text Similarity

Topics to know

- How to do word disambiguation
- Distributed vs distributional representations
- How to compute text similarity
- What word embeddings capture

Main Idea of word2vec

- Predict between every word and its context

- Two algorithms
  - Skip-gram (SG)
    Predict context words given target (position independent)
  - Continuous Bag of Words (CBOW)
    Predict target word from bag-of-words context

Slide adapted from Chris Manning

Training Methods

- Two (moderately efficient) training methods

    Hierarchical softmax

    Negative sampling

  Today: naïve softmax

Slide adapted from Chris Manning

Instead, a **bank** can hold the investments in a custodial account

Context words | center word | context words
2 word window | t | 2 word window

But as agriculture burgeons on the east **bank**, the river will shrink

Context words | center | context
2 word window | t | 2 word window

Objective Function

- Maximize the probability of context words given the center word

$$J'(\Theta) = \prod_{t=1} \prod_{\substack{-m\leq j\leq m \\ j\neq 0}} P(w_{t+j} \mid w_{t\ j}\ \Theta)$$

Negative log likelihood

$$J'(\Theta) = -1/T \sum_{t=1} \sum_{\substack{-m\leq j\leq m \\ j\neq 0}} \log\ P(w_{t+j} \mid w_t\ )$$

Where Θ represents all variables to be optimized

Slide adapted from Chris Manning

- What are the parameters in the objective function? What are we learning?

What are the parameters in the objective function? What are we learning?

Softmax
using word c to obtain probability of word o

- Convert $P(w_{t+j} | w_t)$

$$P(o|c) = \exp(u_o^T \; v_c) / \Sigma^V_{w=1} \exp(u_w^T v_c)$$

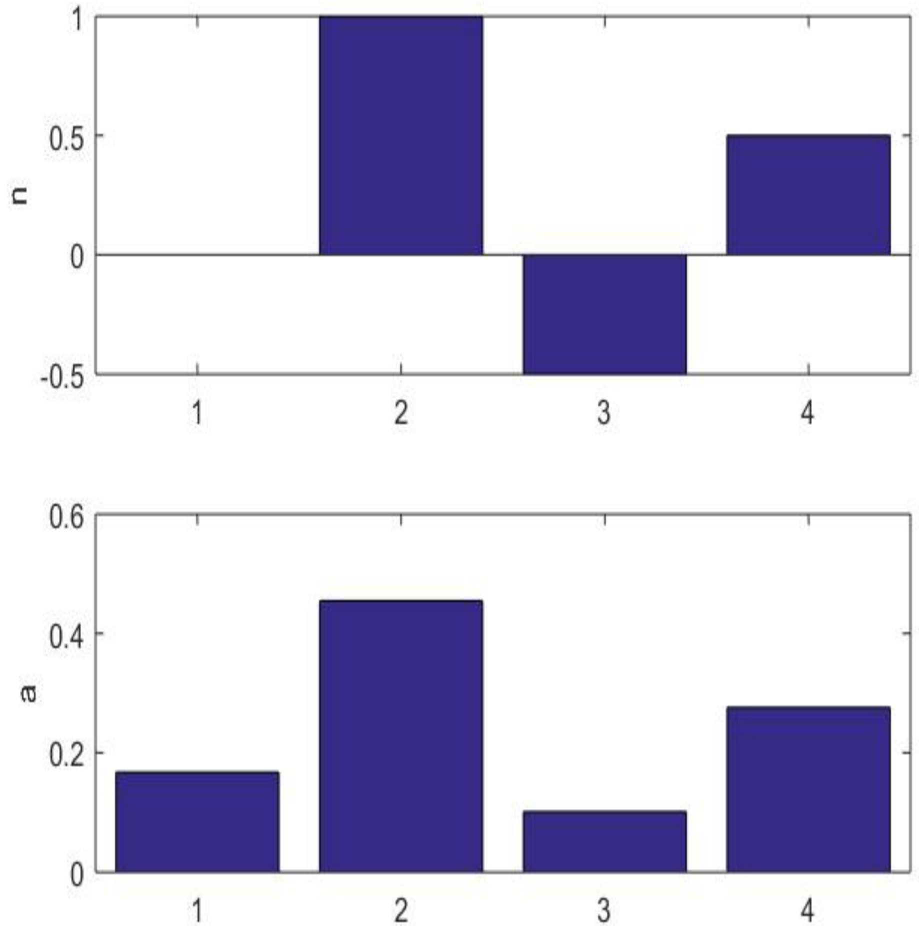
exponentiate normalize

to make positive

where o is the outside (or output) word index and c is the center word index, v_c and u_o are center and outside vectors of indices c and o

Slide adapted from Chris Manning

Softmax



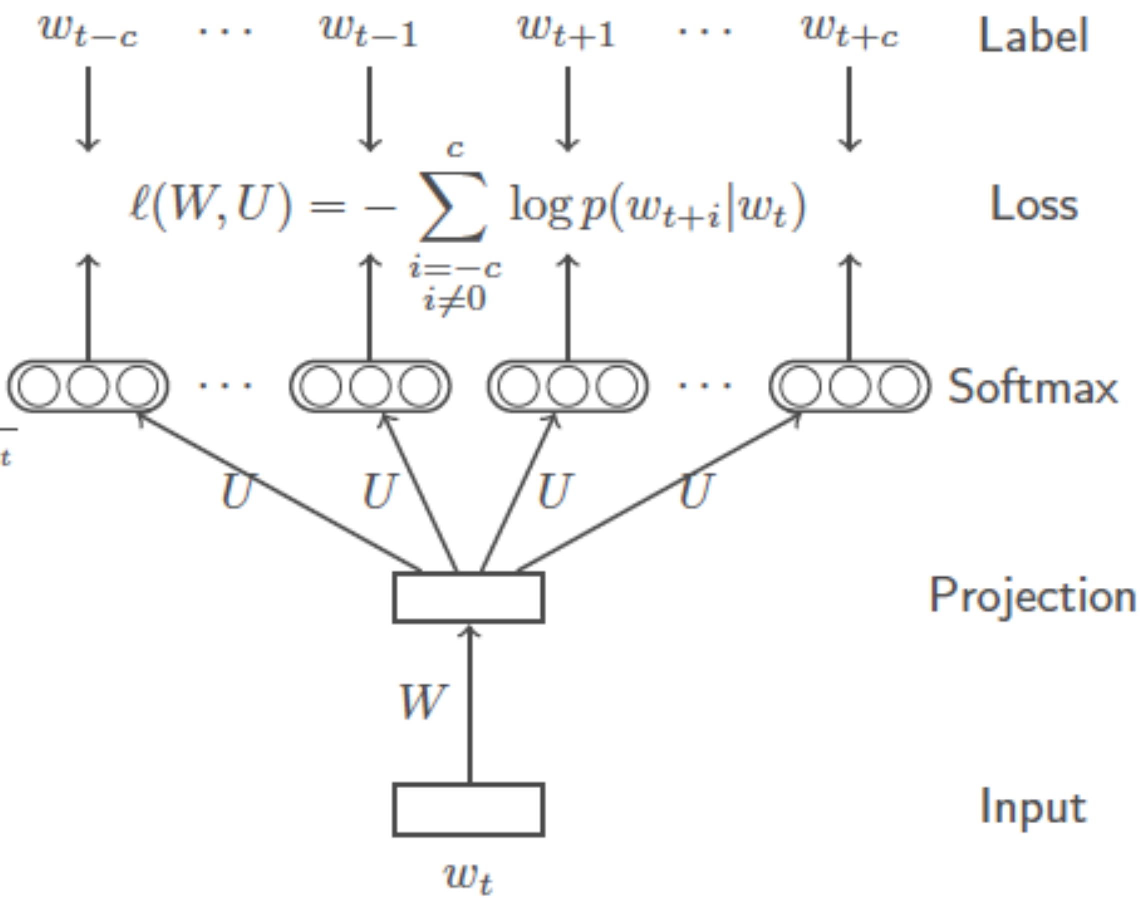
Slide from Dragomir Radev

word2vec

Mikolov et al. (2013)

Skip-gram

- Predict context $w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}$ given target w_t

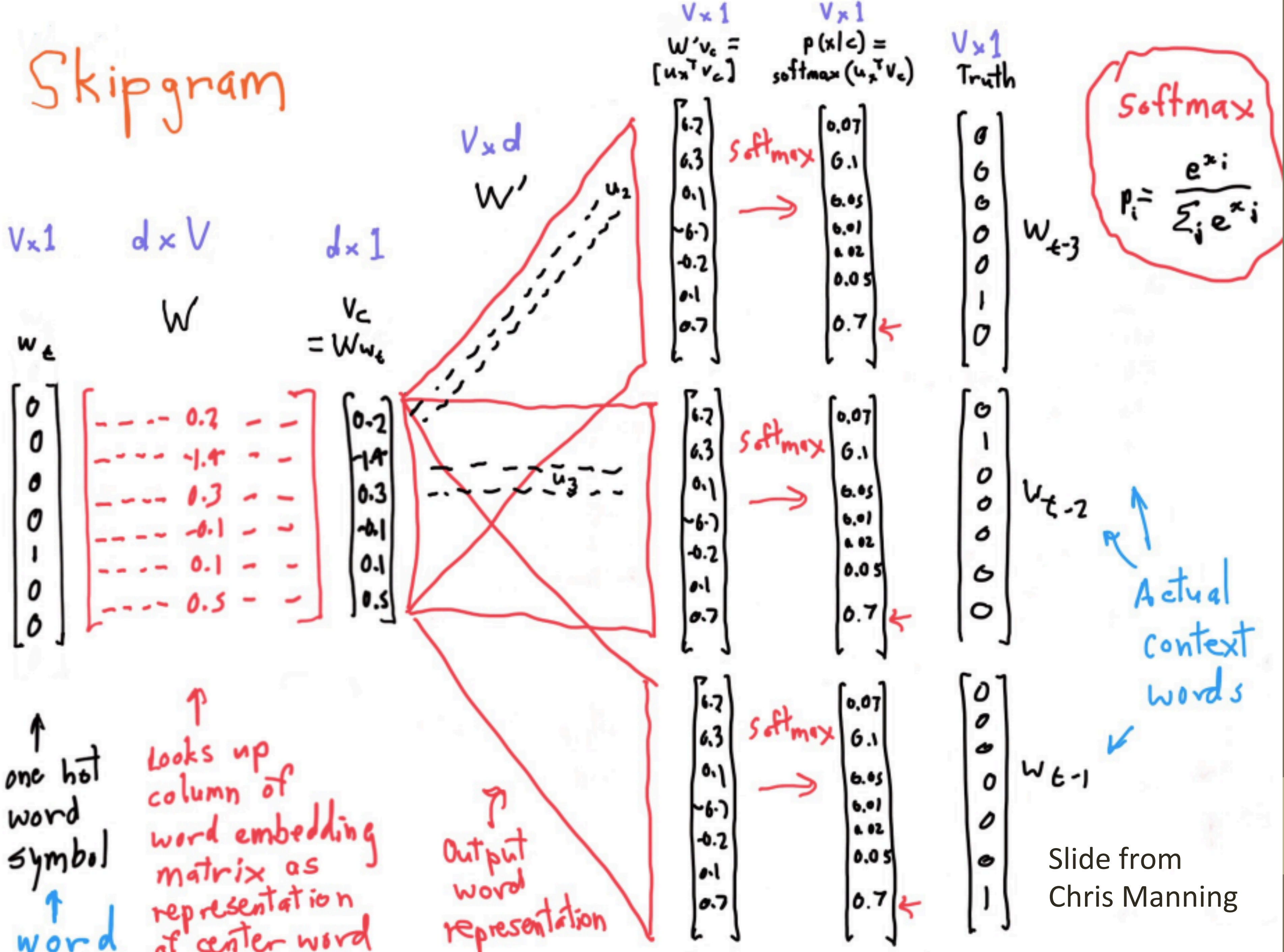


$$p(w_j|w_t) = \frac{e^{U_j h_t}}{\sum_k e^{U_k h_t}}$$

$$h_t = W^\top w_t$$

$$\ell(W,U) = -\sum_{\substack{i=-c \\ i\neq 0}}^{c} \log p(w_{t+i}|w_t)$$

Slide from Kapil Thadani



Slide from Chris Manning

Question

- What are we learning?

- How is the loss computed?

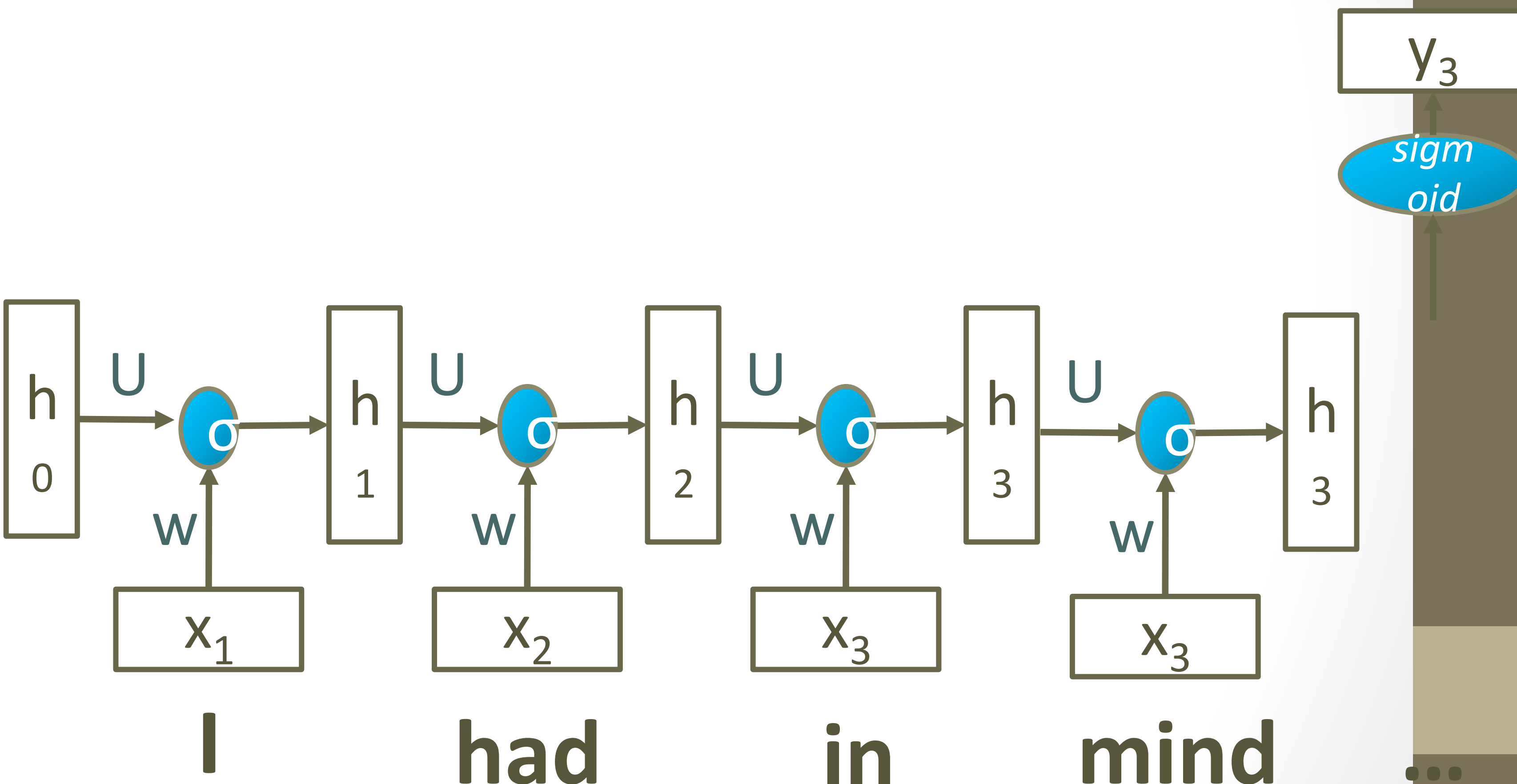
Neural Nets

- Basic architecture of feed forward neural network
- Loss and gradient descent
- Softmax
- Backpropagation
- Determining dimensions of parameters, input and output (basically, all HW3 questions)
- Recurrent Neural Network
- LSTM

Review Session - Bhavana

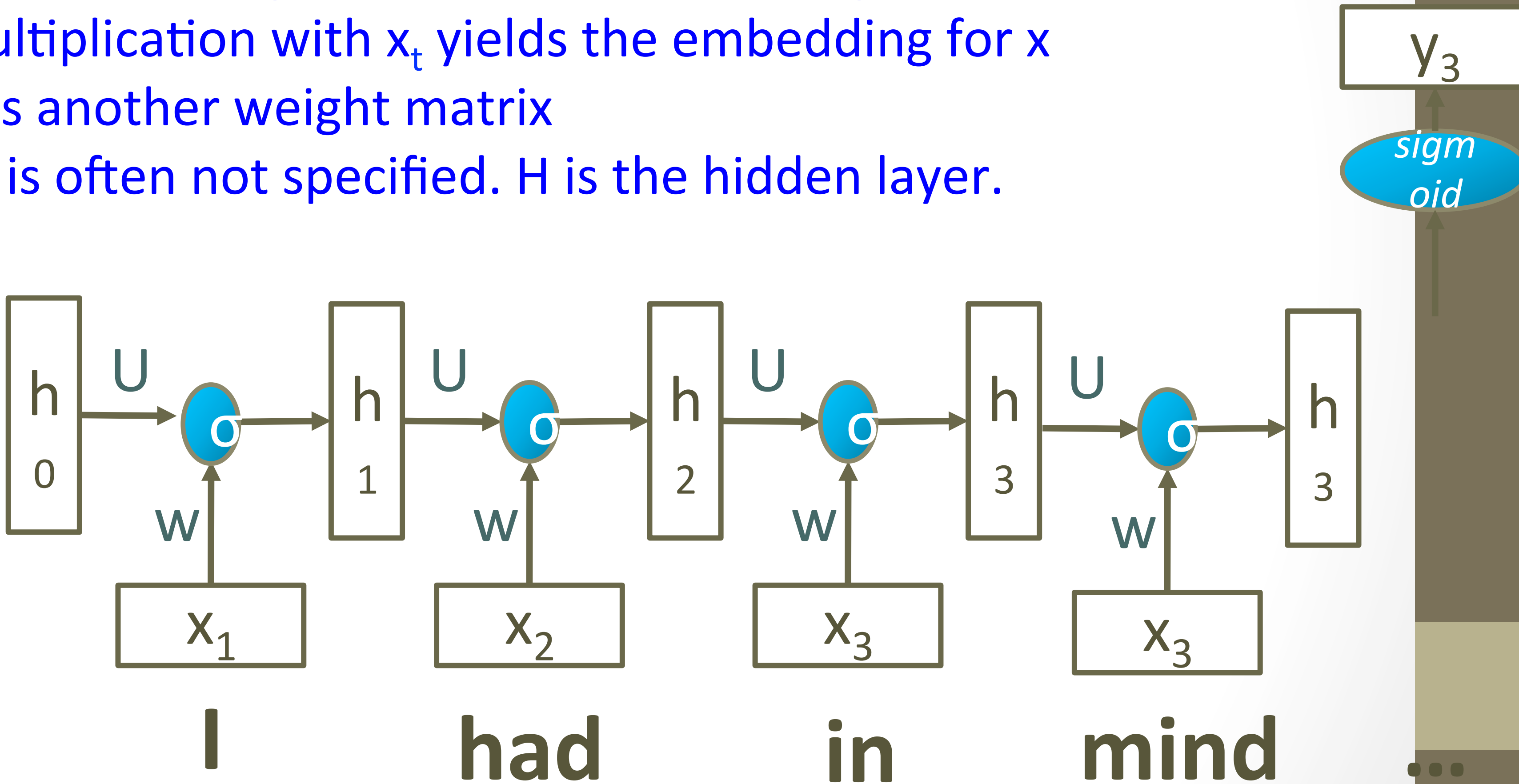
- HW3 answers plus the basics of neural net architectures

RNN – I had in mind your facts, buddy, not hers.



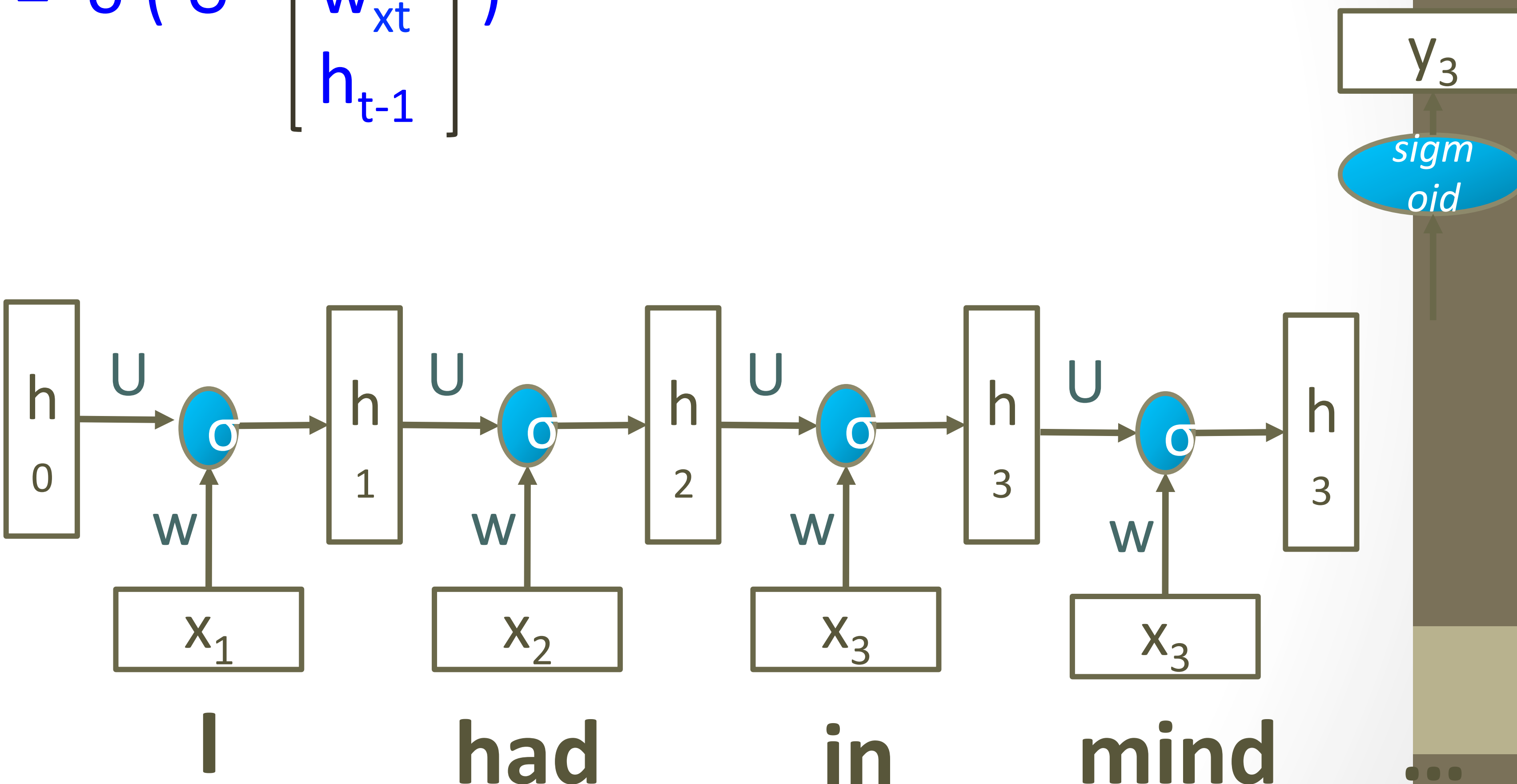
RNN – I had in mind your facts, buddy, not hers.

W are the weights: the word embedding maatrix multiplication with x_t yields the embedding for x
U is another weight matrix
H_0 is often not specified. H is the hidden layer.



RNN – I had in mind your facts, buddy, not hers.

$$h_t = \sigma (U \begin{bmatrix} w_{xt} \\ h_{t-1} \end{bmatrix})$$



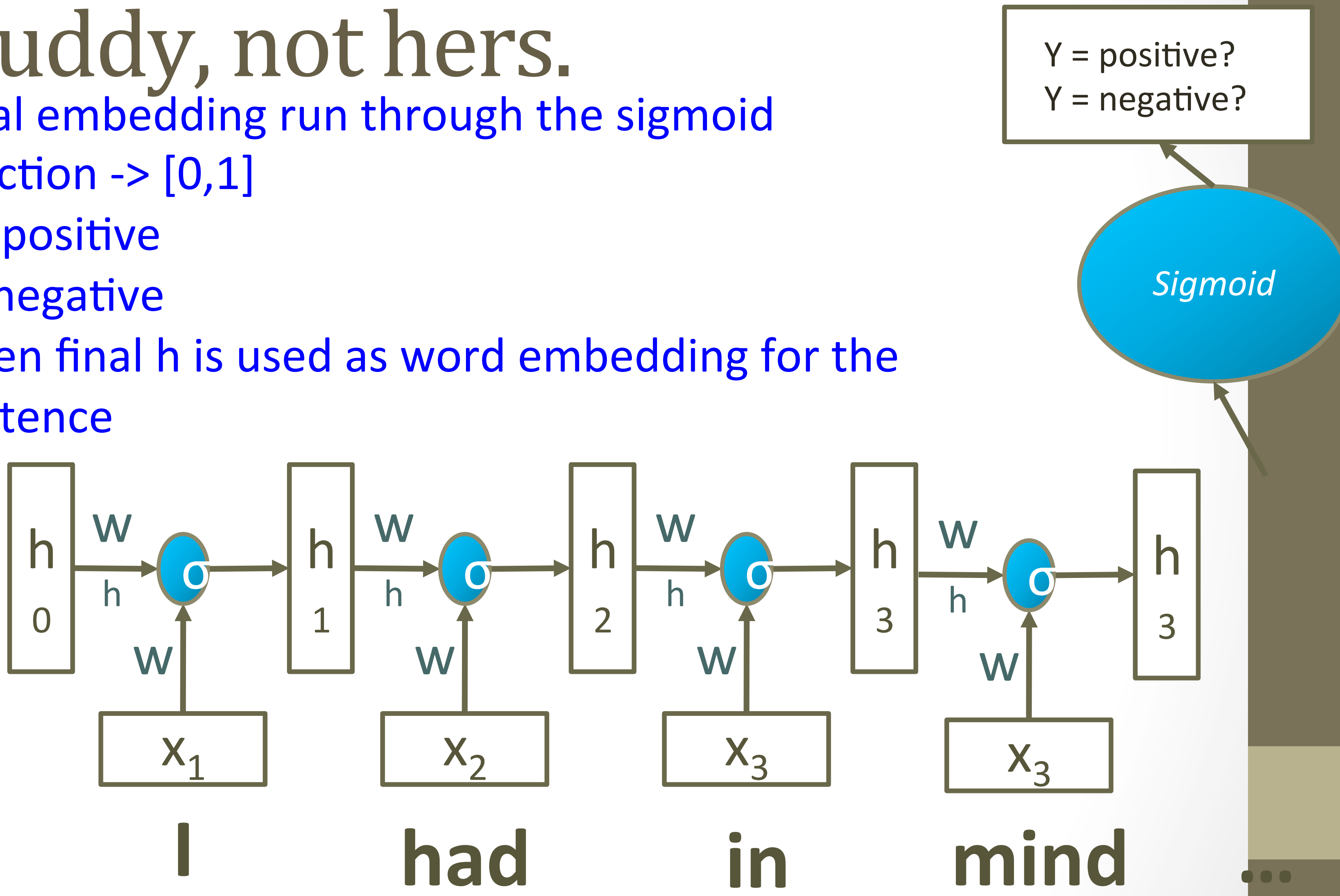
RNN – I had in mind your facts, buddy, not hers.

Final embedding run through the sigmoid function -> [0,1]

1 = positive

0= negative

Often final h is used as word embedding for the sentence



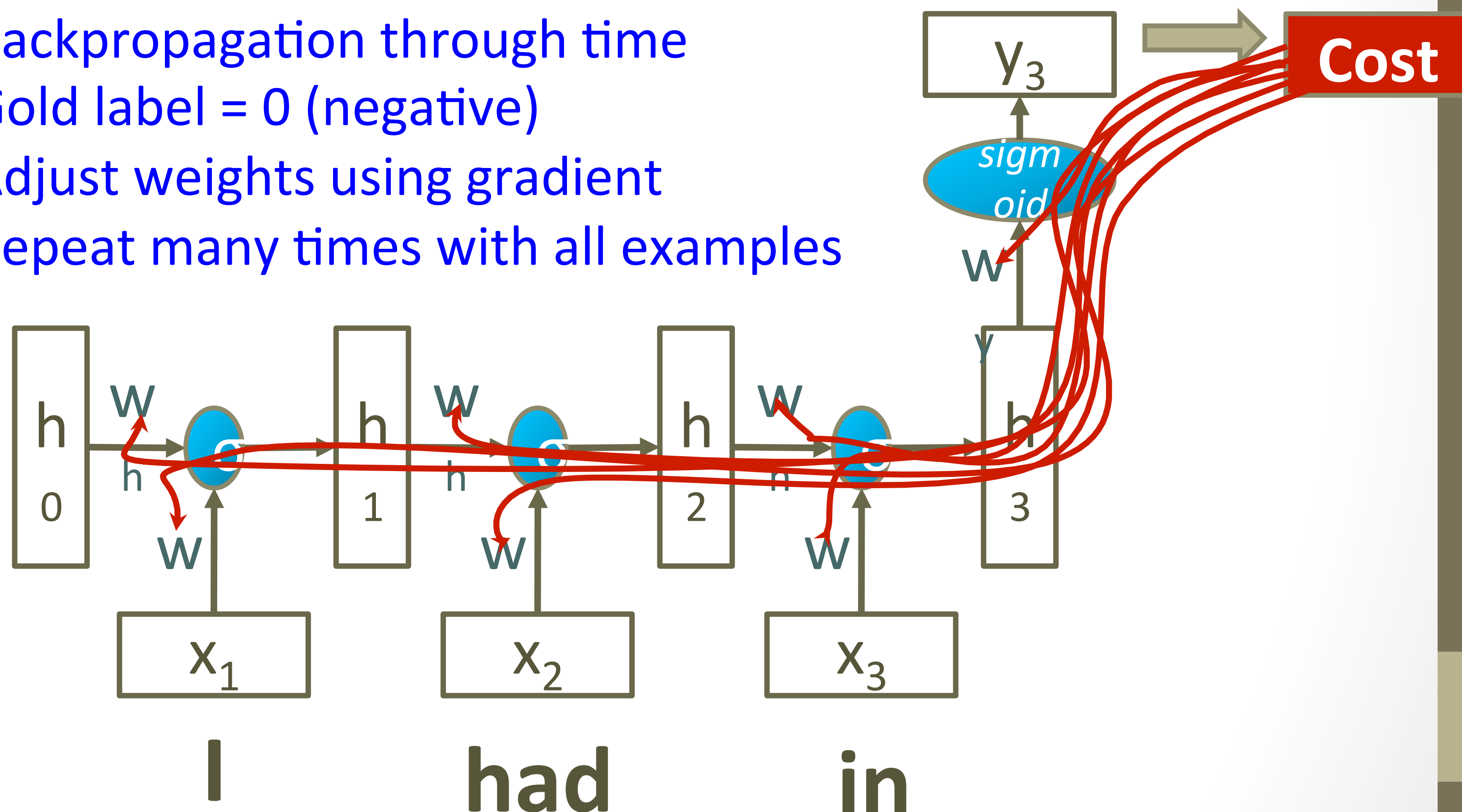
Questions

- How is h computed?
- What parameters are learned?
- How is y predicted ?
- What are the problems with an RNN?

What are the problems with a RNN?

Updating Parameters of an RNN

Backpropagation through time
Gold label = 0 (negative)
Adjust weights using gradient
Repeat many times with all examples



Slide from Radev

Question

Question 30

Suppose you are given the following step function:

```
def step(x_t, h_tm1, c_tm1):
   u_t = T.nnet.sigmoid(T.dot(params[“Wx”], x_t) +
       T.dot(params[“Wh”], h_tm1))

   # Calculate the input gate
   i = T.nnet.sigmoid(T.dot(params[“Wxi”], x_t) +
     T.dot(params[“Whi”], h_tm1))

   # Calculate the forget gate
   f = T.nnet.sigmoid(T.dot(params[“Wxf”], x_t) +
     T.dot(params[“Whf”], h_tm1))

   # Calculate the output gate
   o = T.nnet.sigmoid(T.dot(params[“Wxo”], x_t) +
     T.dot(params[“Who”], h_tm1))

   # Find the memory cell value for the current time
step
   c_t = f * c_tm1 + i * u_t

   # Find the hidden value for the current time step
   h_t = o * T.tanh(c_t)

   return h_t, c_t
```

Assume that

T.nnet.sigmoid applies the sigmoid function.

T.tanh applies the tanh function.

T.dot(A, B) computes the dot product of A and B.

The * operator will perform elementwise multiplication when applied to two vectors.

params is a dictionary of parameters that have already been initialized.

Which deep learning architecture does this function belong to?

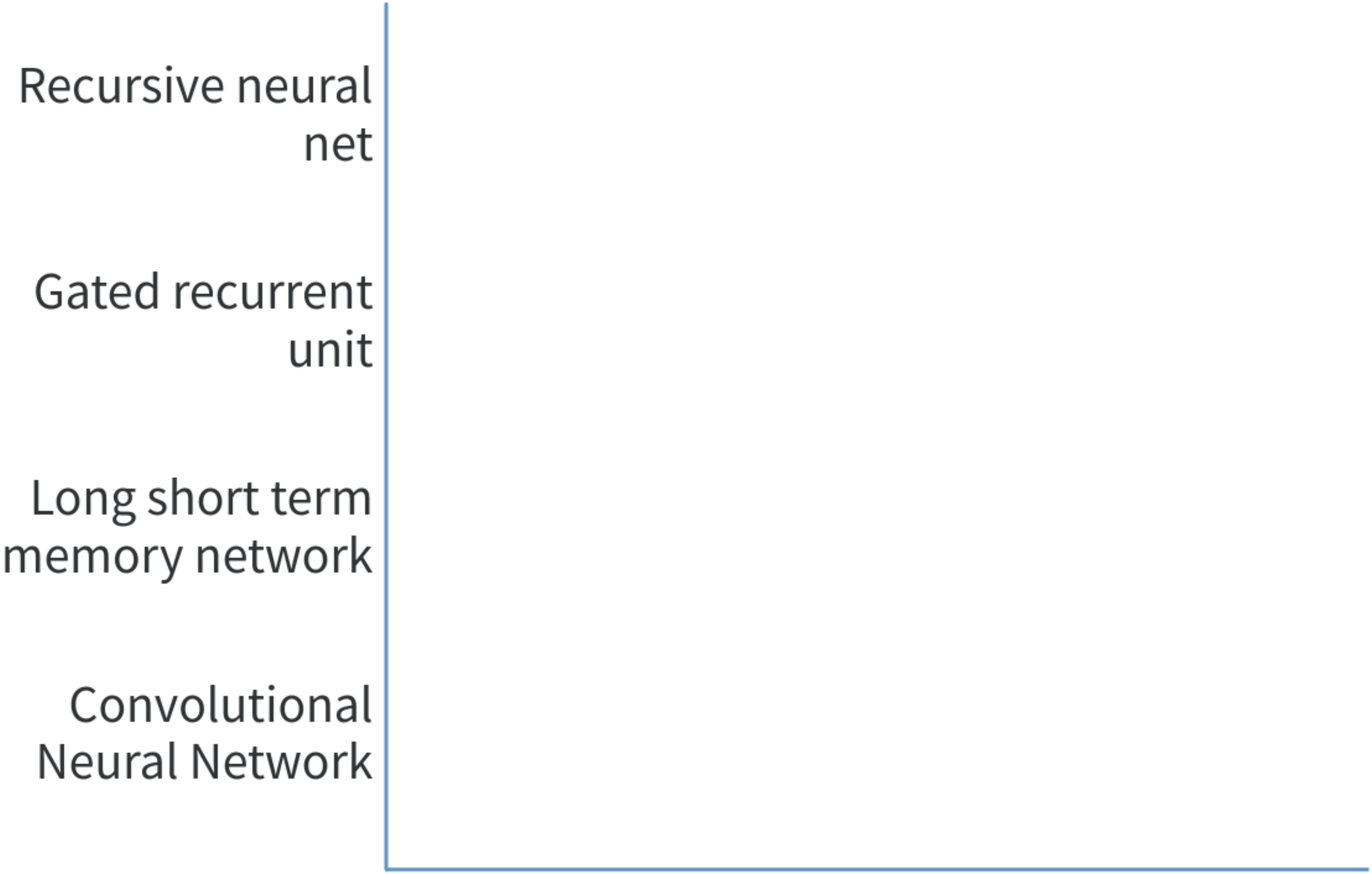
a. Recursive Neural Network

b. Gated Recurrent Unit

c. Long Short Term Memory Network

d. Convolutional Neural Network

Which deep learning architecture does the function belong to?



Gated Architectures

- RNN: at each state of the architecture, the entire memory state (h) is read and written

- Gate = binary vector g ε {0,1}
  - Controls access to n-dimensional vector x•g

- Consider $s' \leftarrow g \odot x + (1 - g) \odot (s)$

  - Reads entries from x specified by g
  - Copies remaining entries from s (or h as we've been labeling the hidden state)

$$
\underset{\mathbf{s}'}{\begin{bmatrix} 8 \\ 11 \\ 3 \\ 7 \\ 5 \\ 15 \end{bmatrix}} \leftarrow \underset{\mathbf{g}}{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}} \odot \underset{\mathbf{x}}{\begin{bmatrix} 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{bmatrix}} + \underset{(\mathbf{1}-\mathbf{g})}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \odot \underset{\mathbf{s}}{\begin{bmatrix} 8 \\ 9 \\ 3 \\ 7 \\ 5 \\ 8 \end{bmatrix}}
$$

Example: gate copies from positions 2 and 5 in the input
Remaining elements copied from memory

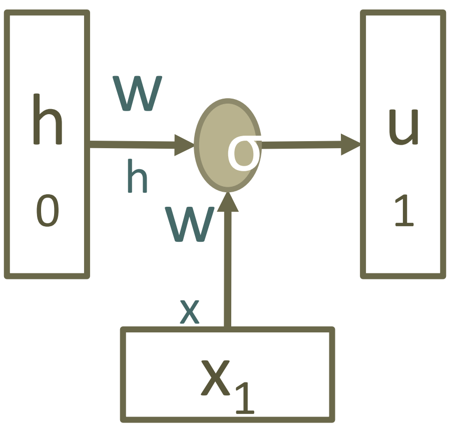
LSTM Solution

- Use memory cell to store information at each time step.
- Use “gates” to control the flow of information through the network.
  - Input gate: protect the current step from irrelevant inputs
  - Output gate: prevent the current step from passing irrelevant outputs to later steps
  - Forget gate: limit information passed from one cell to the next

[slides from Catherine Finegan-Dollak]

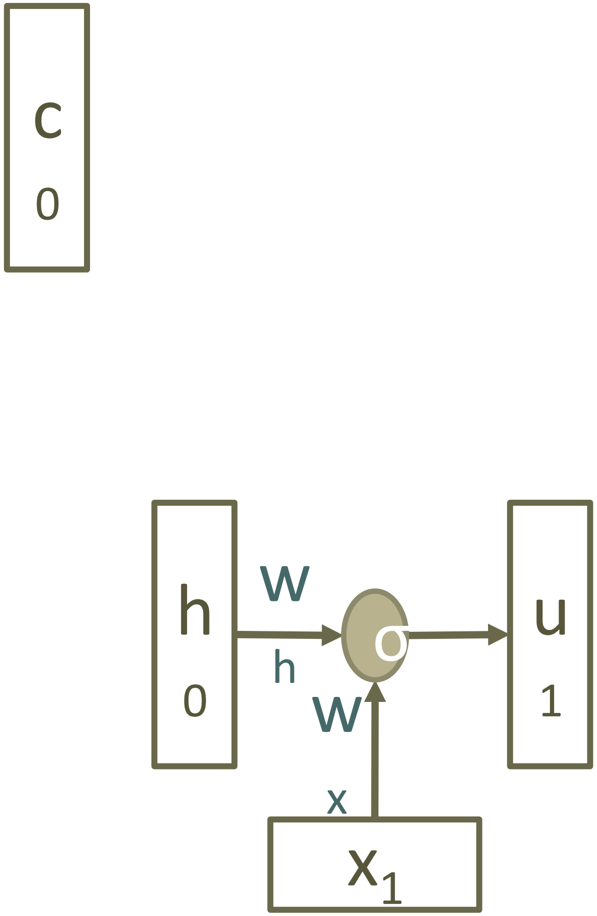
Transforming RNN to LSTM

$$u_t = \sigma(W_h h_{t-1} + W_x x_t)$$



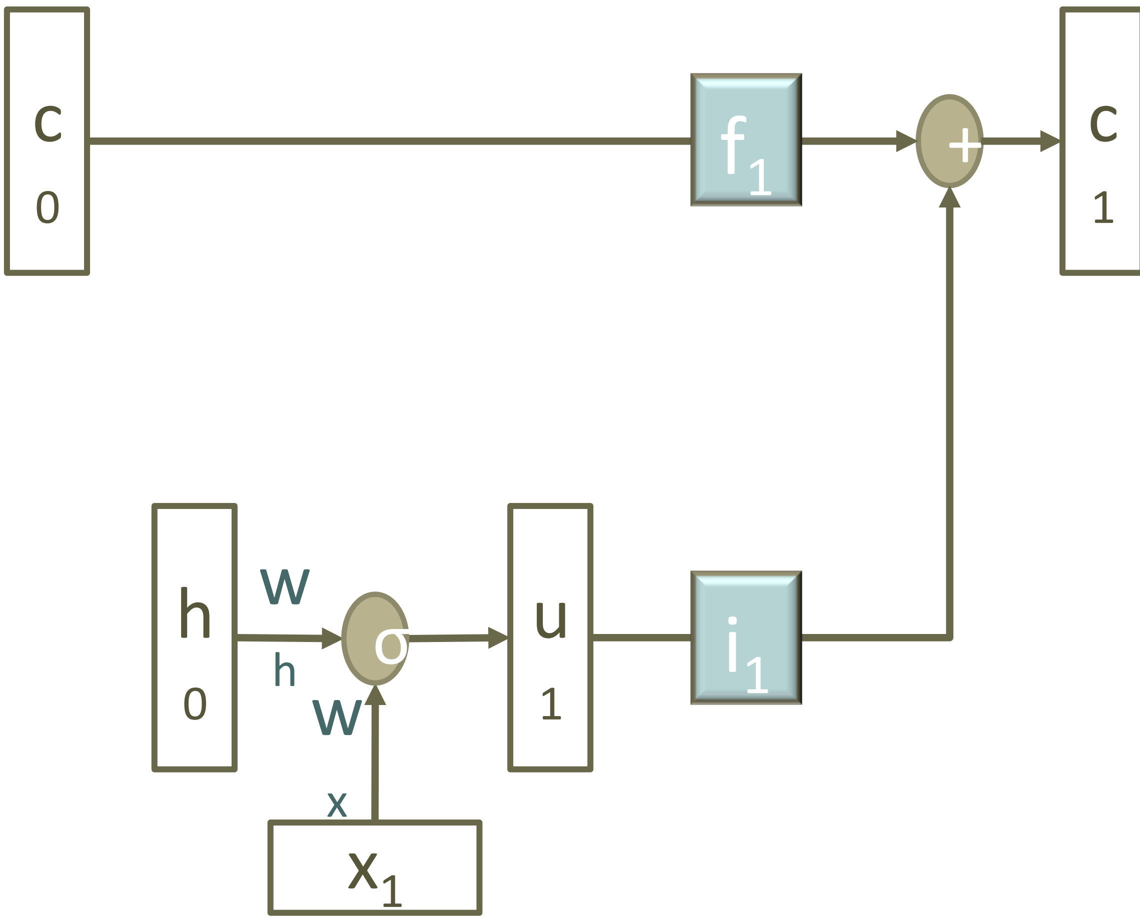
[slides from Catherine Finegan-Dollak]

Transforming RNN to LSTM



[slides from Catherine Finegan-Dollak]

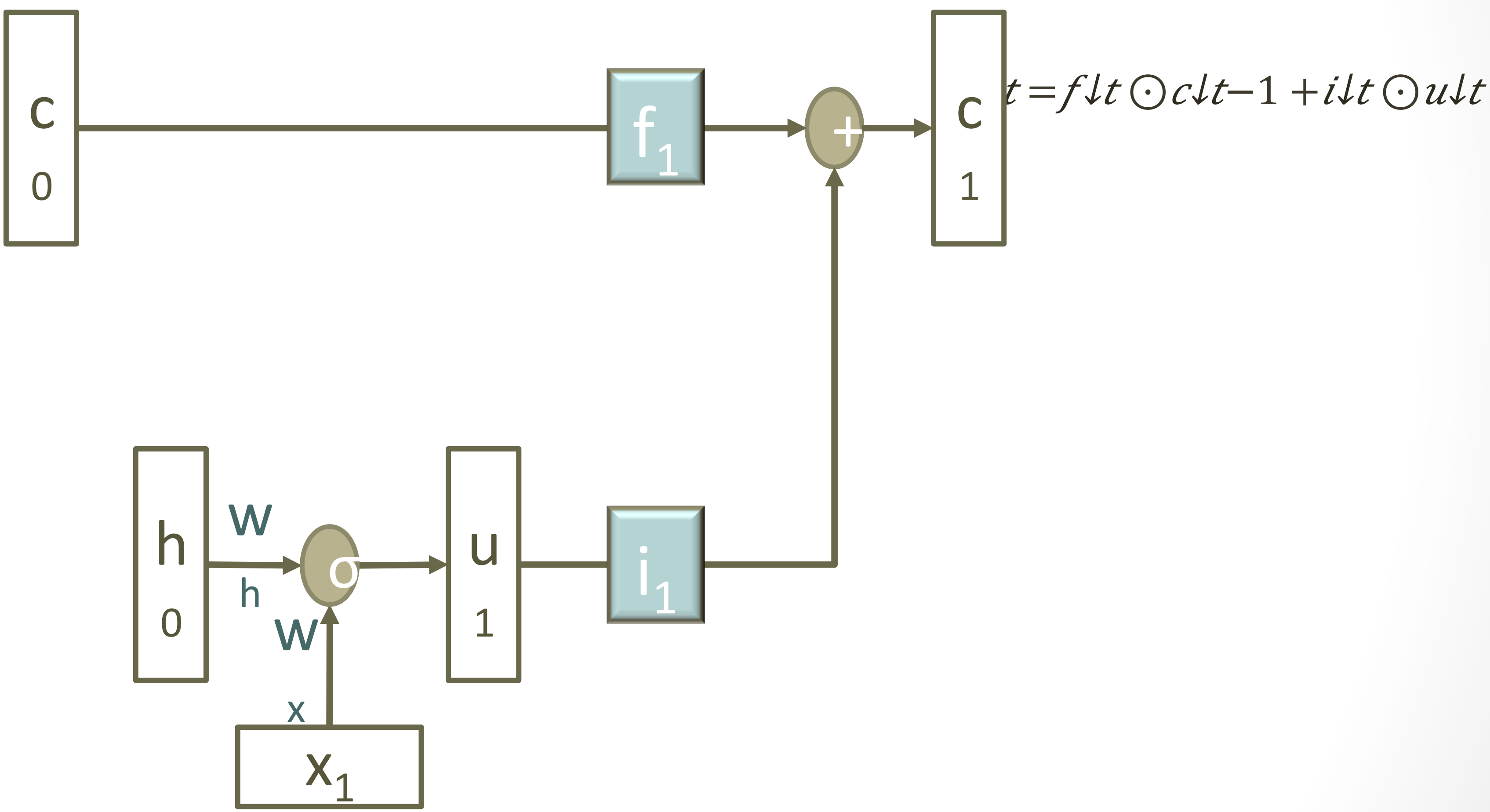
Transforming RNN to LSTM



$t = f_t \odot c_{t-1} + i_t \odot u_t$

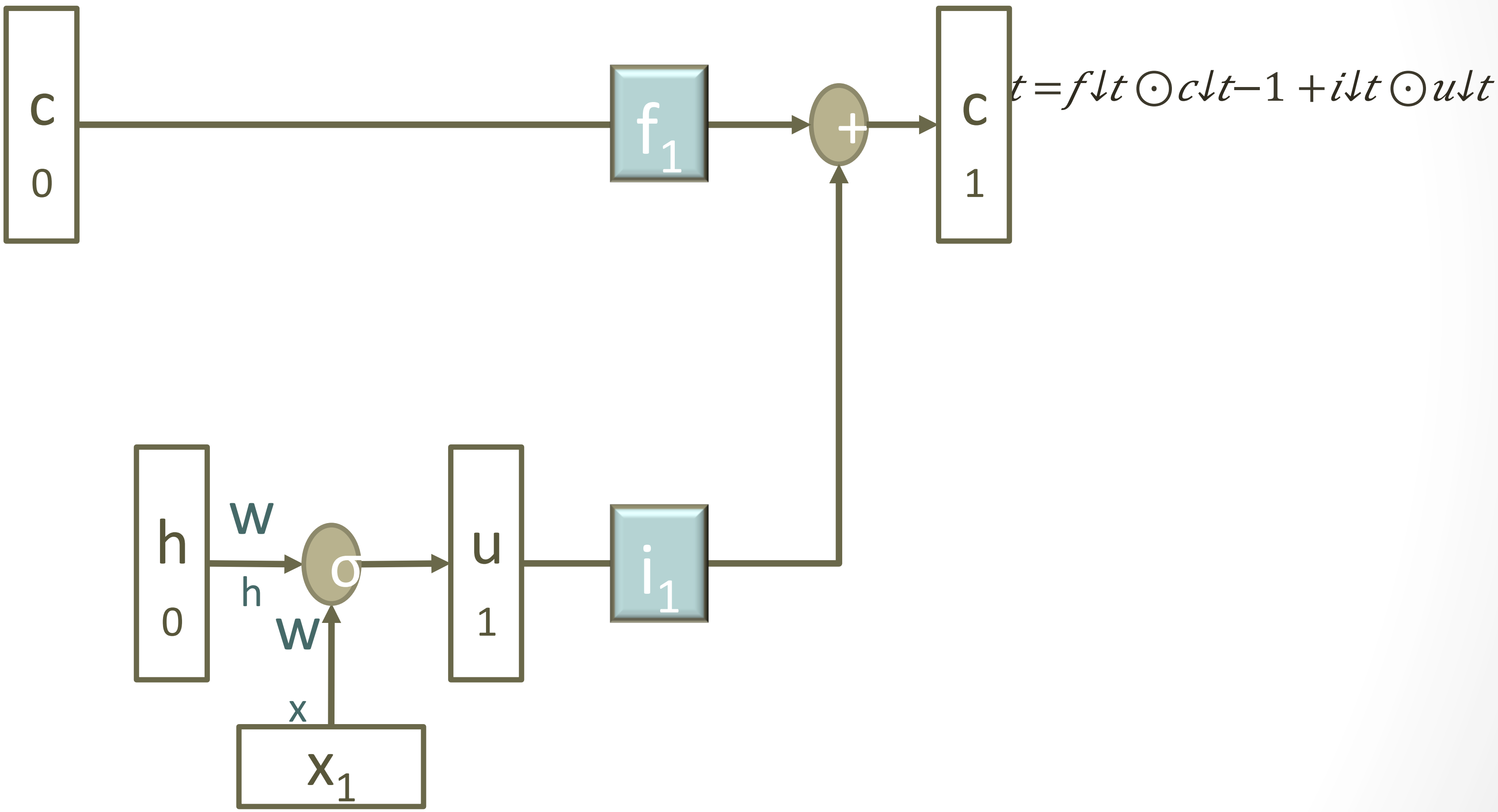
[slides from Catherine Finegan-Dollak]

Transforming RNN to LSTM



[slides from Catherine Finegan-Dollak]

Transforming RNN to LSTM



[slides from Catherine Finegan-Dollak]

Summarization

- Extractive vs abstractive summarization
- Indicative vs informative summary
- Single document vs multi-document
- Generic vs user-focused

Extraction methods

- Topic signature words
- Graph-based methods

Topic Signature Words

- Uses the log ratio test to find words that are highly descriptive of the input
- the log-likelihood ratio test provides a way of setting a threshold to divide all words in the input into either descriptive or not
  - the probability of a word in the input is the same as in the background
  - the word has a different, higher probability, in the input than in the background
- Binomial distribution used to compute the ratio of the two likelihoods
- The sentences containing the highest proportion of topic signatures are extracted.

Log likelihood ratio

$$\lambda = \frac{b(k, N, p)}{b(k_I, N_I, p_I).b(k_B, N_B, p_B)}$$

Where the counts with subscript i occur in the input corpus and those with subscript B occur in the background corpus

Probability (p) of w occuring k times in N Bernoulli trials

The statistic -2λ has a known statistical distribution: chi-squared

Graph-based methods

- Sentence similarity is measured as a function of word overlap
  - Frequently occurring words link many sentences
  - Similar sentences give support for each other's importance
- Input represented as highly connectived graph
  - Vertices represent sentences
  - Edges between sentences weighted by similarity between two sentences
  - Cosine similarity with TF*IDF weights for words

Sentence Selection

- Vertex importance (centrality) computed using graph algorithms
  - Edge weights normalized to form probability distribution -> Markov chain
  - Compute probablity of of being in each vertex of graph at time t while making consecutive transitions from one vertex to next
  - As more transitions made, probability of each vertex converges -> stationary distribution
- Vertices with higher probability = more important sentences

Abstractive summarization

- What is compression?
- What is fusion?
- What traditional method might I use for a supervised compression system?

Dataset for compression (~3000 sentence pairs)

Clarke & Lapata (2008)

Input

- Italian air force fighters scrambled to intercept a Libyan airliner flying towards Europe yesterday as the United Nations imposed sanctions on Libya for the first time in Col Muammar Gaddafi 's turbulent 22 years in power .

Compression

- Italian air force fighters scrambled to intercept a Libyan airliner as the United Nations imposed sanctions on Libya .

Text to Text Generation

Model text transformation as a *structured prediction* problem

- Input: One or more sentences with parses
- Output: Single sentence + parse

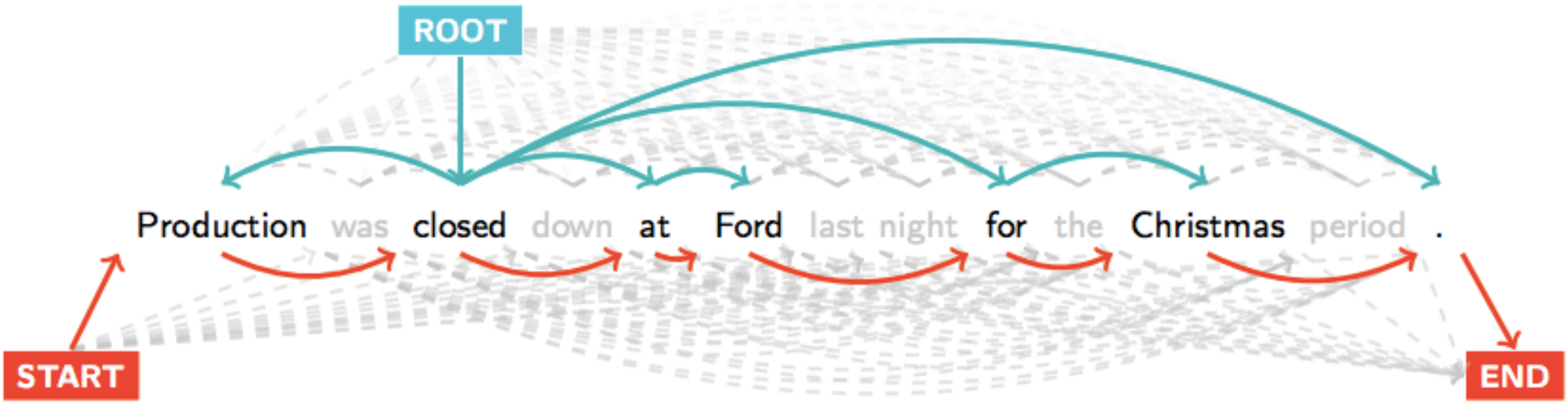
Joint inference over

- **word choice**,
- **n-gram ordering**
- **dependency structure**

Thadani & McKeown, CONLL 2013

structural factorizations

this work



Goal: recover tokens **x**, n-gram sequence **y** and dependency structure **z**

Slide from Thadani

joint inference via ILP

objective

$$
\begin{aligned}
C = \arg\max_{\mathbf{x},\mathbf{y},\mathbf{z}} \; & \boxed{\sum_i x_i \cdot \mathbf{w}_{tok}^{\top} \phi(t_i)} && \text{token score} \\
+ \; & \boxed{\sum_{i,j,k} y_{ijk} \cdot \mathbf{w}_{ngr}^{\top} \phi(\langle t_i, t_j, t_k \rangle)} && \text{ngram score} \\
+ \; & \boxed{\sum_{i,j} z_{ij} \cdot \mathbf{w}_{dep}^{\top} \phi(\langle t_i, t_j \rangle)} && \text{dep score}
\end{aligned}
$$

Slide from Thadani

joint inference via ILP

objective

$$
\begin{aligned}
C = \arg\max_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \sum_i x_i \cdot \mathbf{w}_{tok}^\top \phi(t_i) && \text{token score} \\
+ \quad & \sum_{i,j,k} y_{ijk} \cdot \mathbf{w}_{ngr}^\top \phi(\langle t_i, t_j, t_k \rangle) && \text{ngram score} \\
+ \quad & \sum_{i,j} z_{ij} \cdot \mathbf{w}_{dep}^\top \phi(\langle t_i, t_j \rangle) && \text{dep score}
\end{aligned}
$$

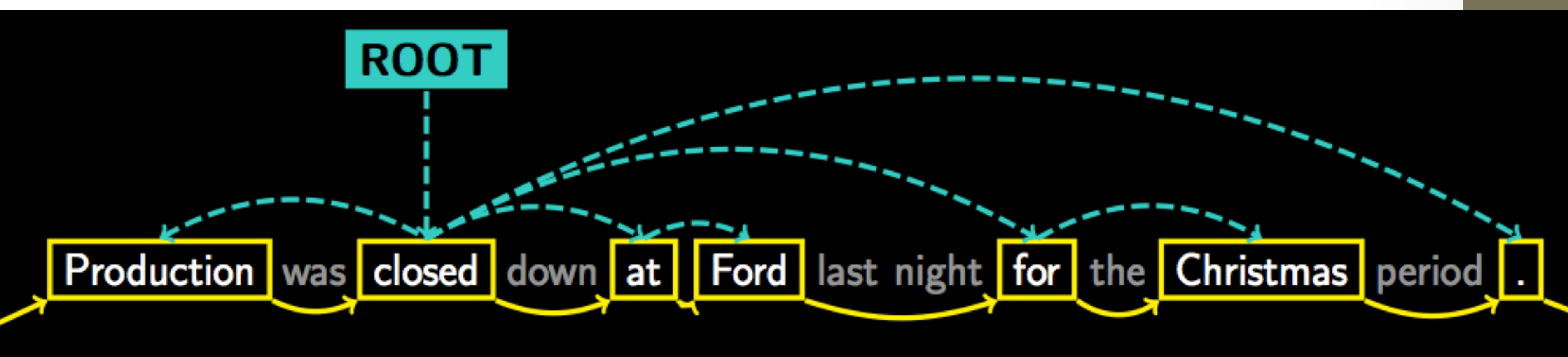
features

- informativeness
- fluency
- fidelity
- pseudo-normalization

Slide from Thadani

Compression

- Input: single sentence
- Output: sentence with **salient** information
- Dataset + baseline from *Clarke & Lapata (2008)*



What about compression using neural nets?

- Dataset: Daily mail highlights

Neural Summarization Architecture

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

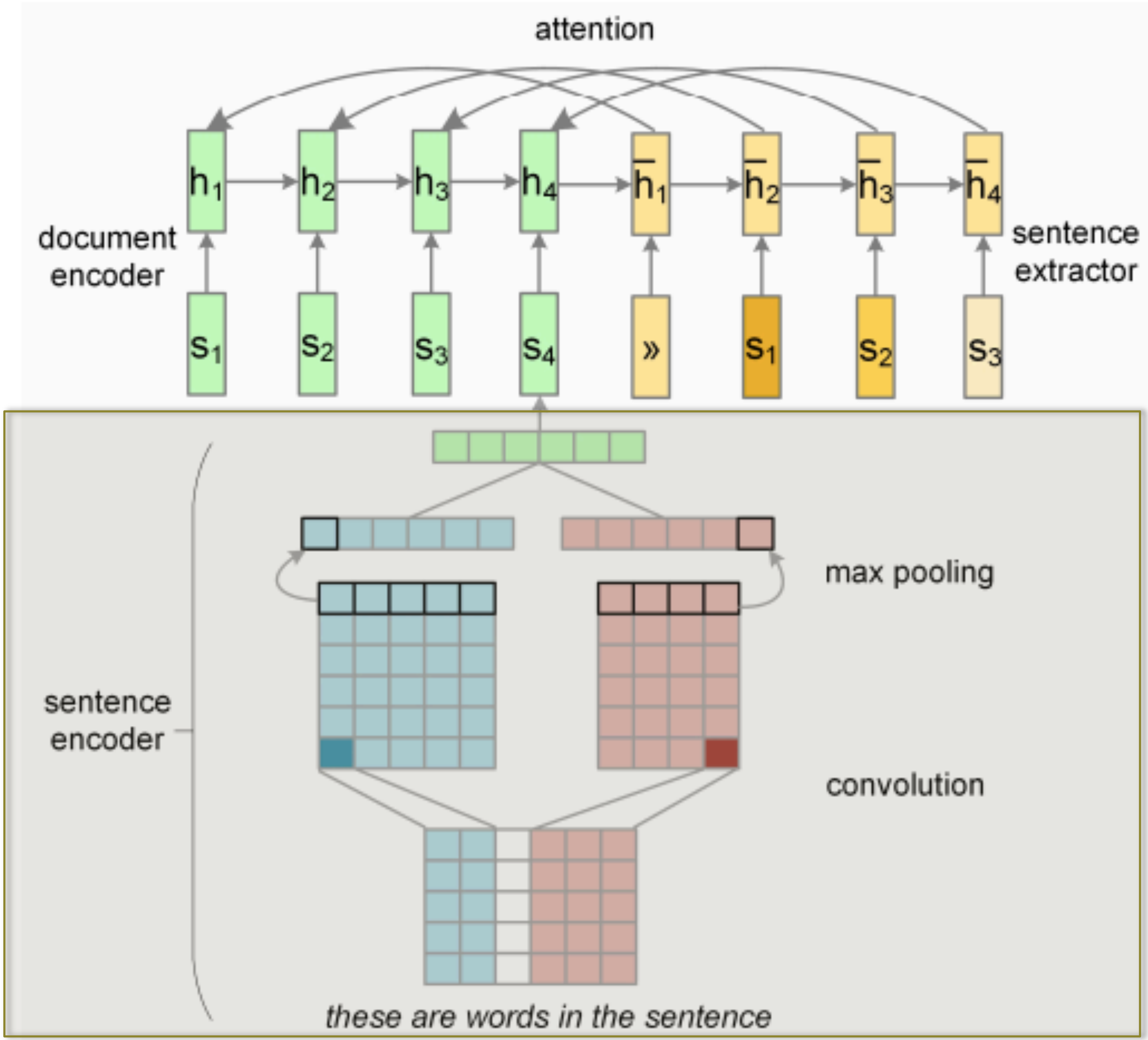
- Attention based hierarchical content extractor

- Encoder-decoder architecture

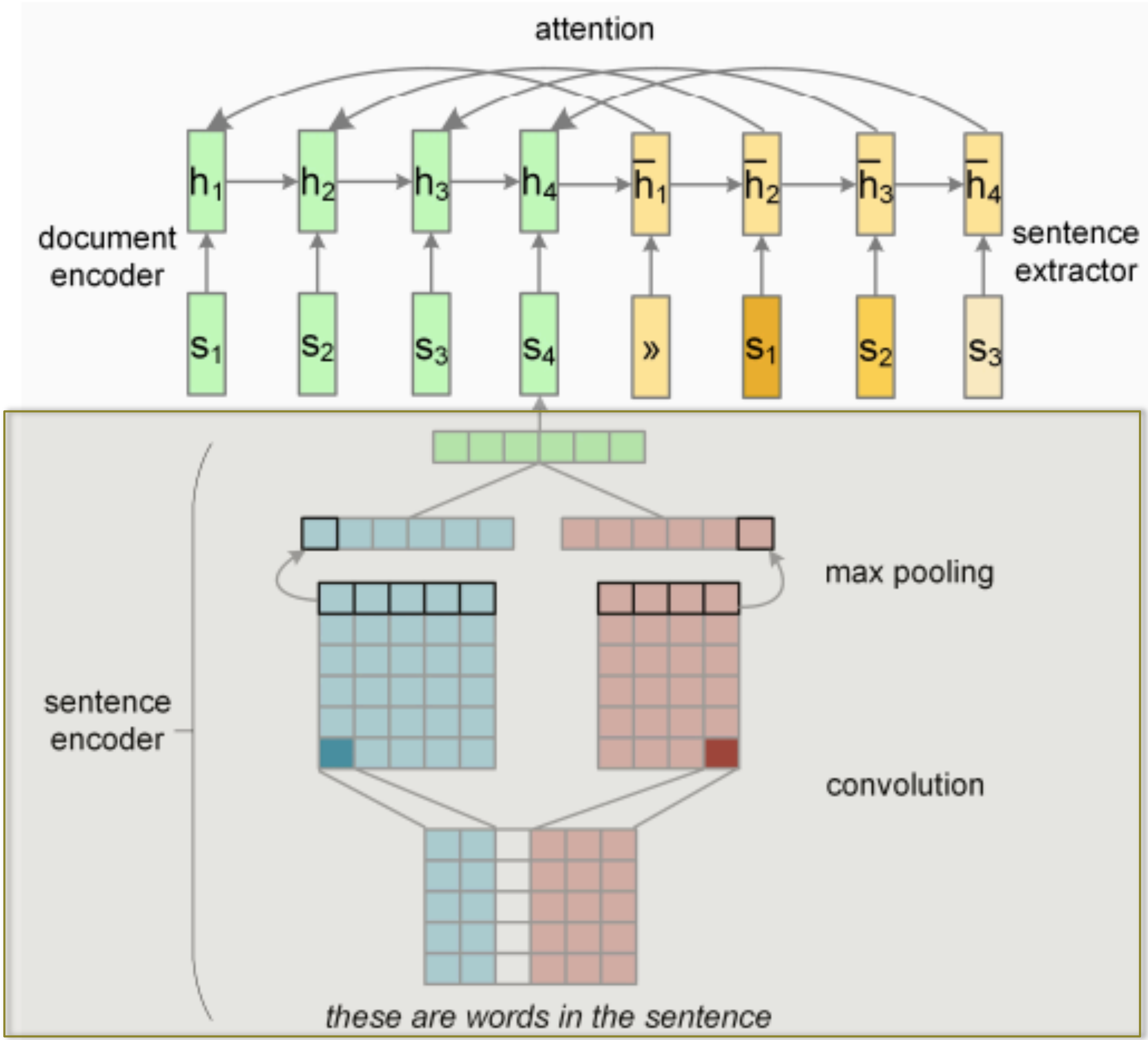
Document Reader

- CNN sentence encoder
  - Useful for sentence classification
  - Easy to train

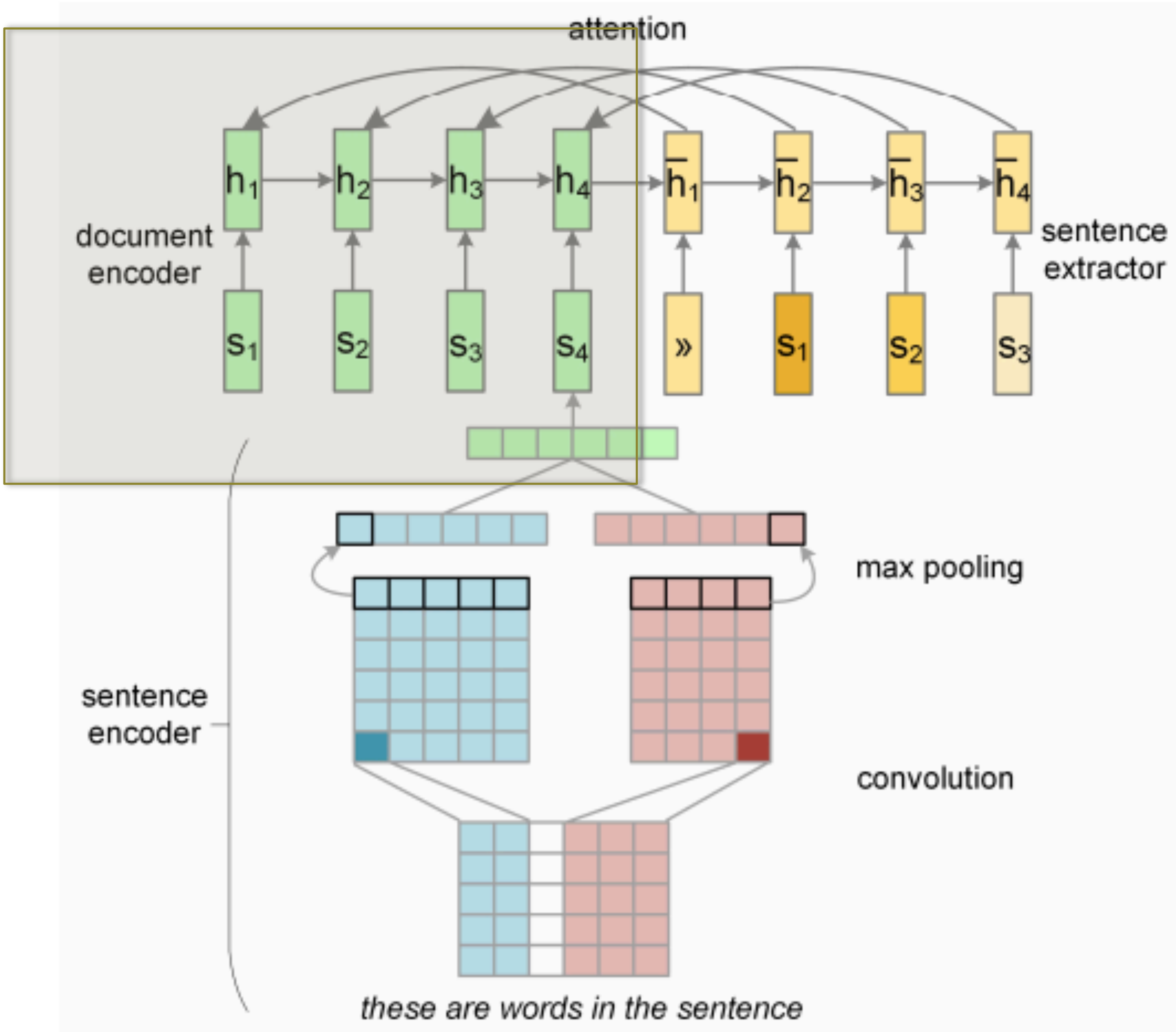
- LSTM document encoder
  - Avoids vanishing gradients



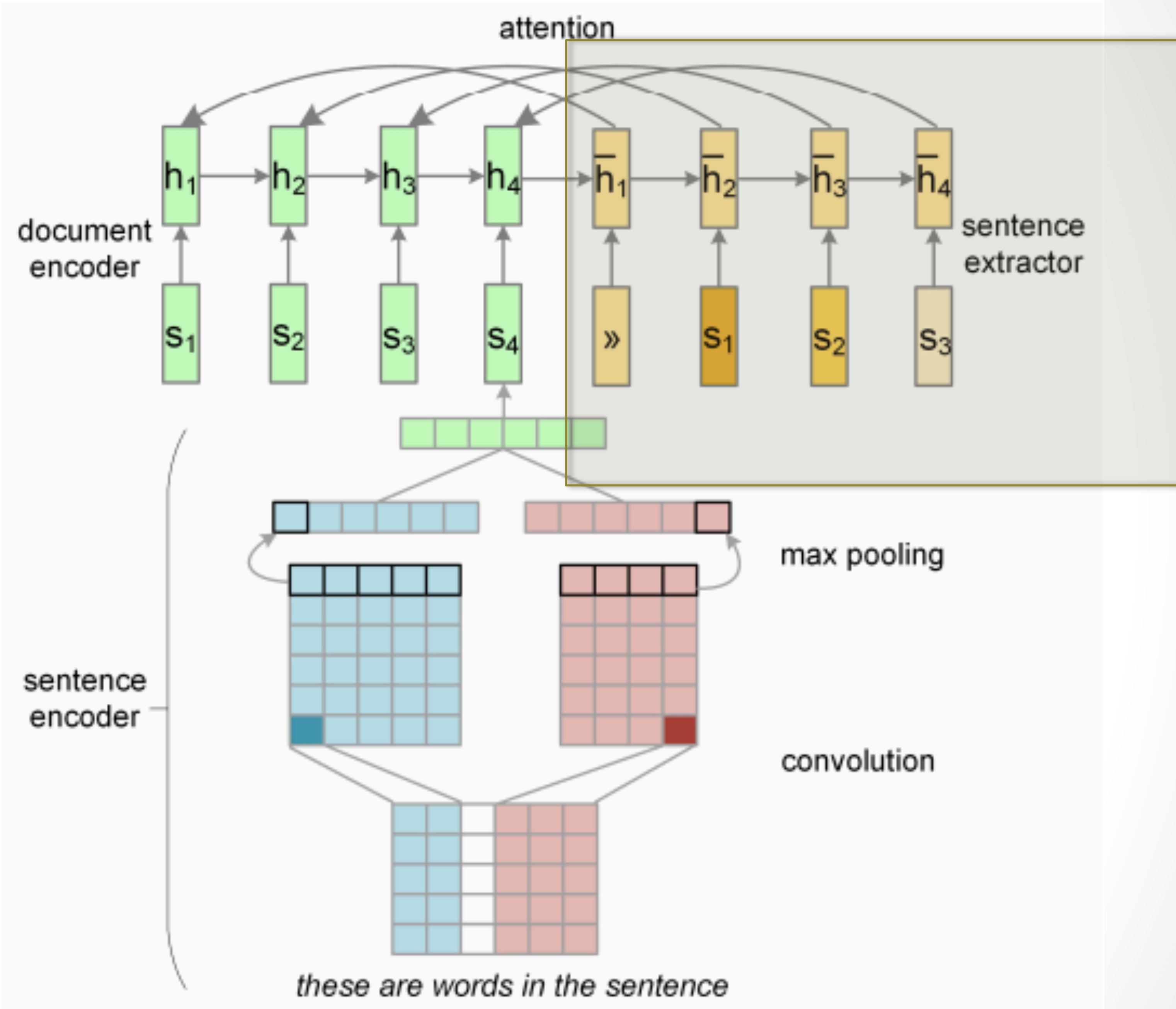
Cheng and Lapata 2016



Cheng and Lapata 2016



Cheng and Lapata 2016



Cheng and Lapata 2016

Types of summarization evaluation

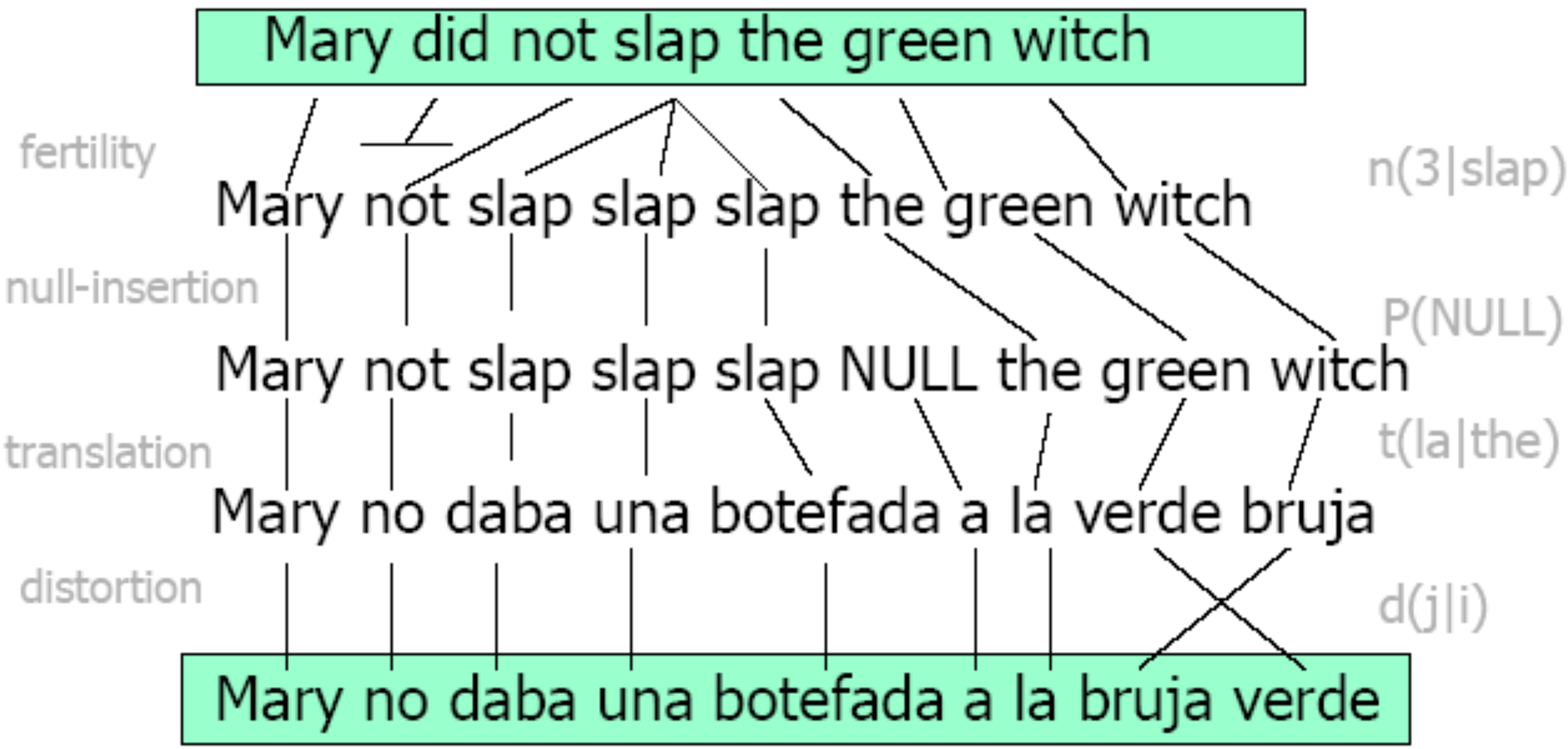
- Automated: Rouge scores
- Manual: Pyramid scores
  - What are they?
- Task based evaluation
  - Does a summary help you to perform a research task better?

Machine Translation

- Challenges for multilingual translation
- What is the MT pyramid?
- What are the different trained models used in the IBM model?
- What is phrased-based MT?

Statistical MT

IBM Model (Word-based Model)



http://www.clsp.jhu.edu/ws03/preworkshop/lecture_yamada.pd

IBM's EM trained models (1-5)

- Word translation
- Local alignment
- Fertilities
- Class-based alignment
- Re-ordering

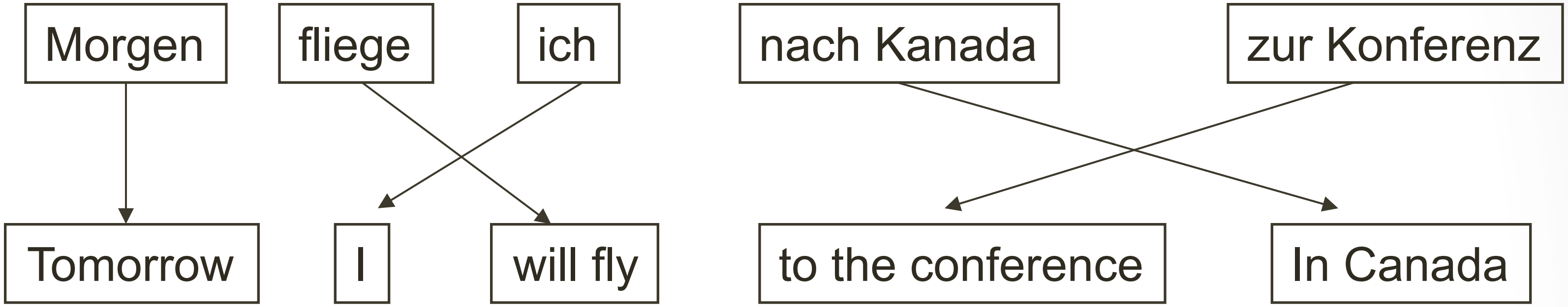
All are separate models to train!

Model 1:

$$p(f,a \mid e) = p(a \mid e) * p(f \mid a,e) = \frac{c}{(n+1)^m} \prod_{j=1}^{m} p(f_j \mid e_{a_j})$$

Slide courtesy of Kevin Knight http://www.sims.berkeley.edu/courses/is290-2/f04/lectures/mt-lecture.ppt

Phrase-Based Statistical MT



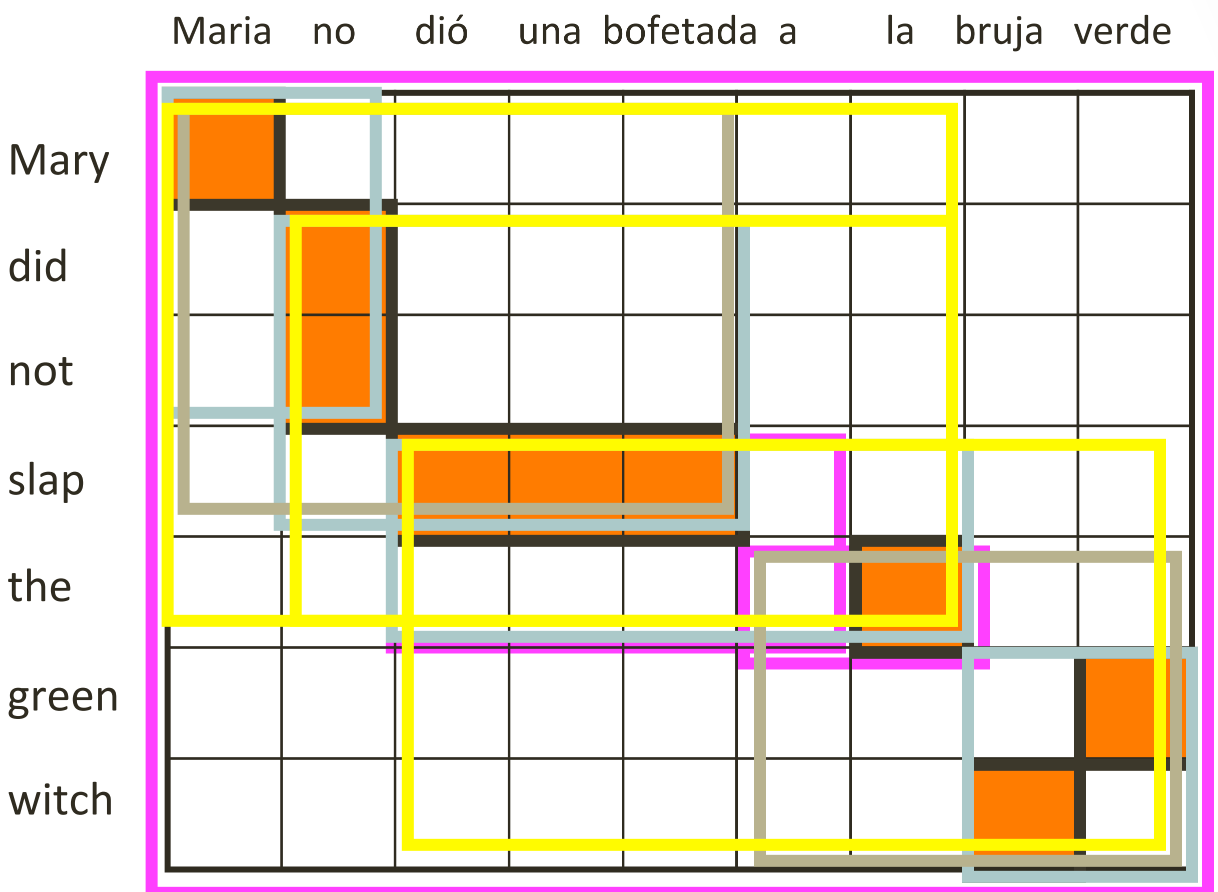
- Foreign input segmented in to phrases
  - "phrase" is any sequence of words
- Each phrase is probabilistically translated into English
  - P(to the conference | zur Konferenz)
  - P(into the meeting | zur Konferenz)
- Phrases are probabilistically re-ordered

See [Koehn et al, 2003] for an intro.

This was state-of-the-art before neural MT

Slide courtesy of Kevin Knight http://www.sims.berkeley.edu/courses/is290-2/f04/lectures/mt-lecture.ppt

Word Alignment Induced Phrases



(Maria, Mary) (no, did not) (slap, dió una bofetada) (la, the) (bruja, witch) (verde, green)

(a la, the) (dió una bofetada a, slap the)

(Maria no, Mary did not) (no dió una bofetada, did not slap), (dió una bofetada a la, slap the)

(bruja verde, green witch) (Maria no dió una bofetada, Mary did not slap)

(a la bruja verde, the green witch) …

(Maria no dió una bofetada a la bruja verde, Mary did not slap the green witch)

How is MT evaluation done?

- Automated metrics: Bleu, Meteor
- Human judgments:
  - Adequacy (accuracy)
  - Fluency
- How were human judgments done in WMT 2017?
  - What were some approaches to quality control for crowd sourcing?

When did Neural MT surpass statistical methods (phrase-based and syntax)?

- WMT 2016

- When did companies first release NMT systems?
  - 2016

Neural MT

- Encoder-decoder approach
- What is the problem with a basic RNN?
- How is attention used?
- How else has the RNN memory problem been addressed?

What other approaches?

- Train stacked RNNS using multiple layers
- Use a bidirectional encoder
  - This can help in remembering the early part of the source input sentence
- Train the input sequence in reverse order
- Deeper networks: decoder depth of 8
- Data: parallel, back-translated, duplicated monolingual

Questions?

Thank you!

- It was great getting to know you!

  Good luck on the exam!