Natural Language Processing

Lecture 16:
Neural Machine Translation,
Attention and Self-attention

12/11/2019

COMS W4705
Yassine Benajiba

Neural Machine Translation

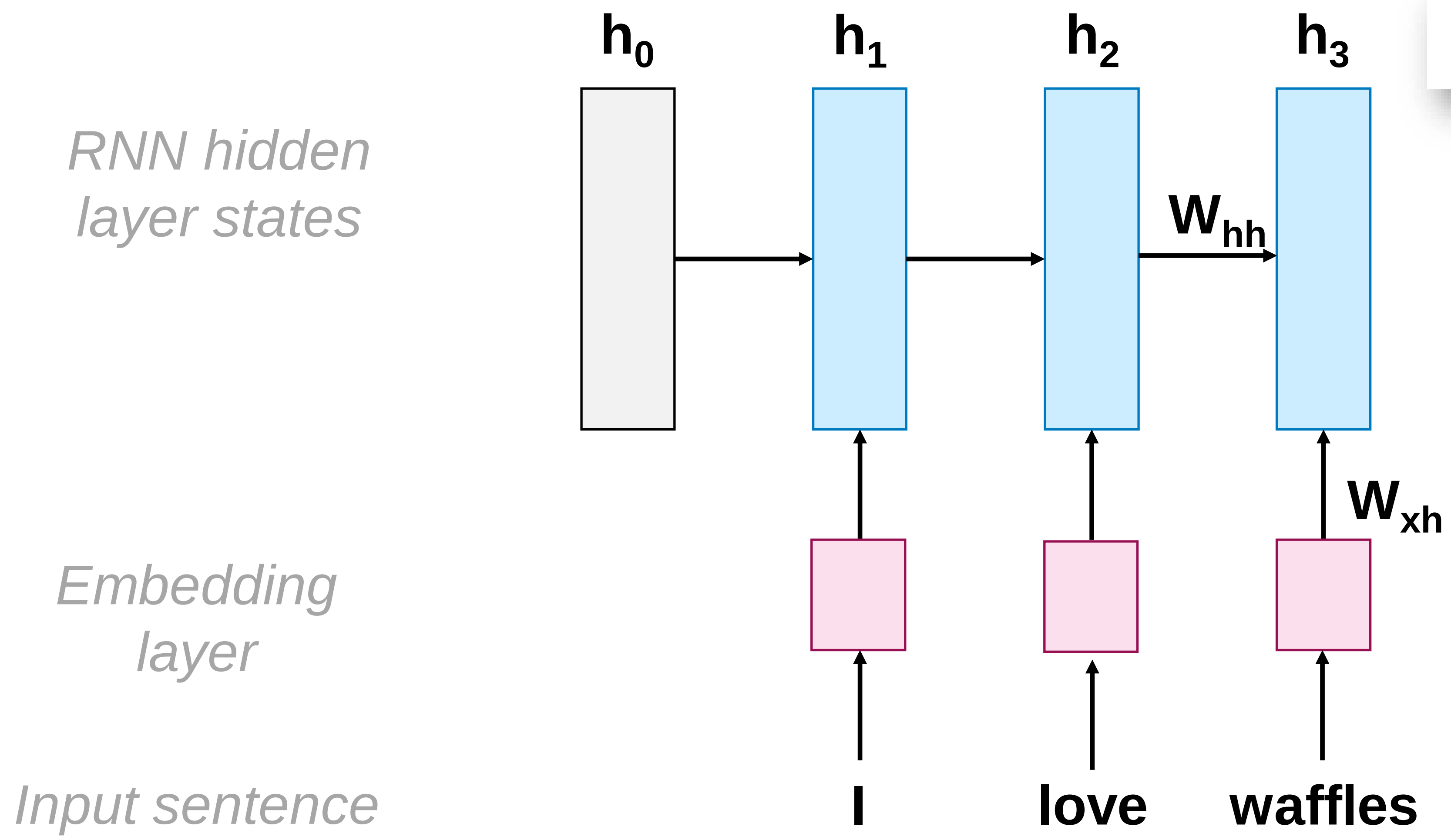
- NMT offers an end-to-end solution for MT

- Uses a sequence-to-sequence (seq2seq) model based on recurrent neural networks

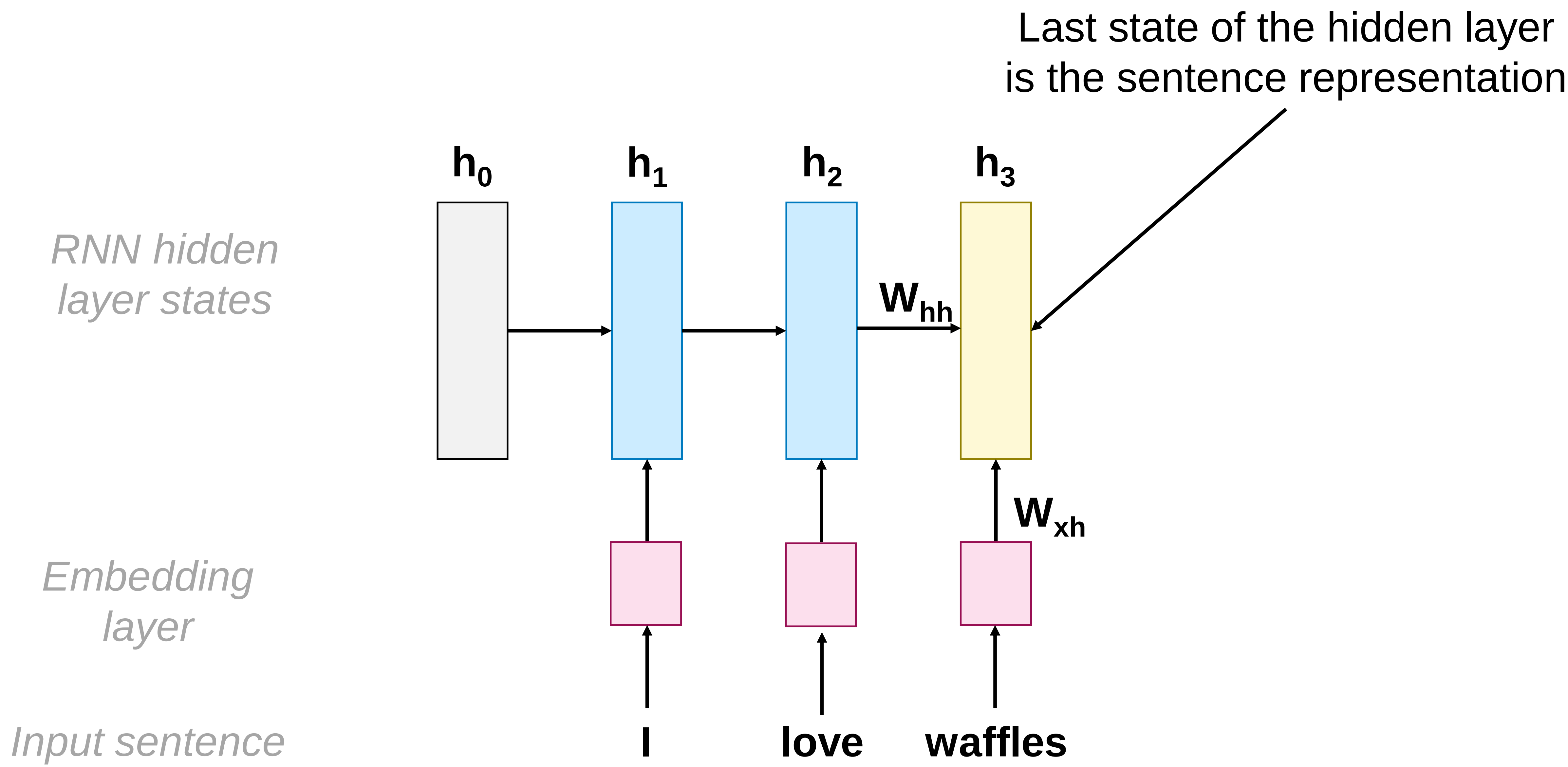
- Simple: no need to build any alignments or phrase tables, model P(E|F) directly.

RNNs refresher

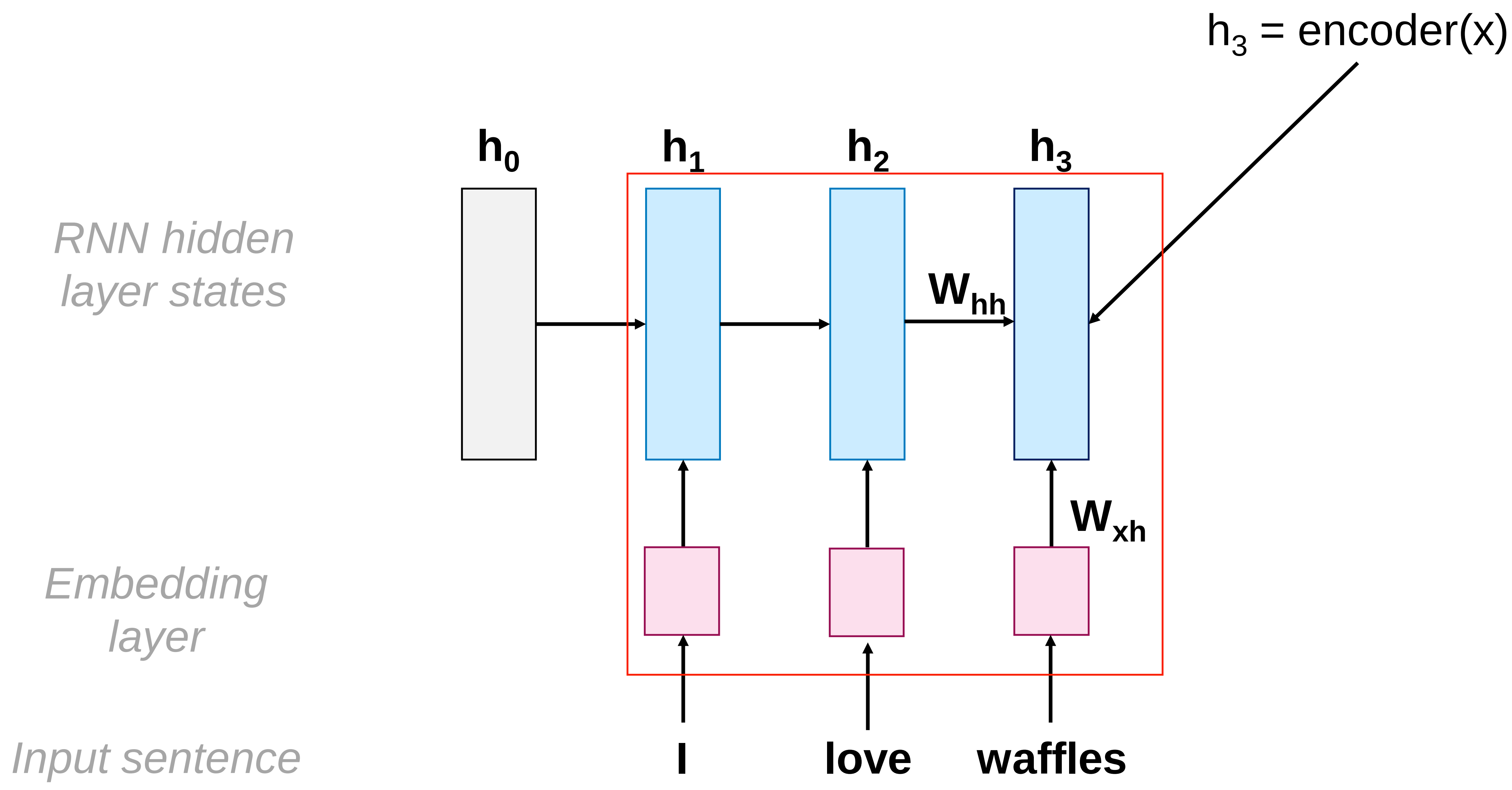
$$h_t = \sigma\left(W_{xh}x_t + W_{hh}h_{t-1}\right)$$



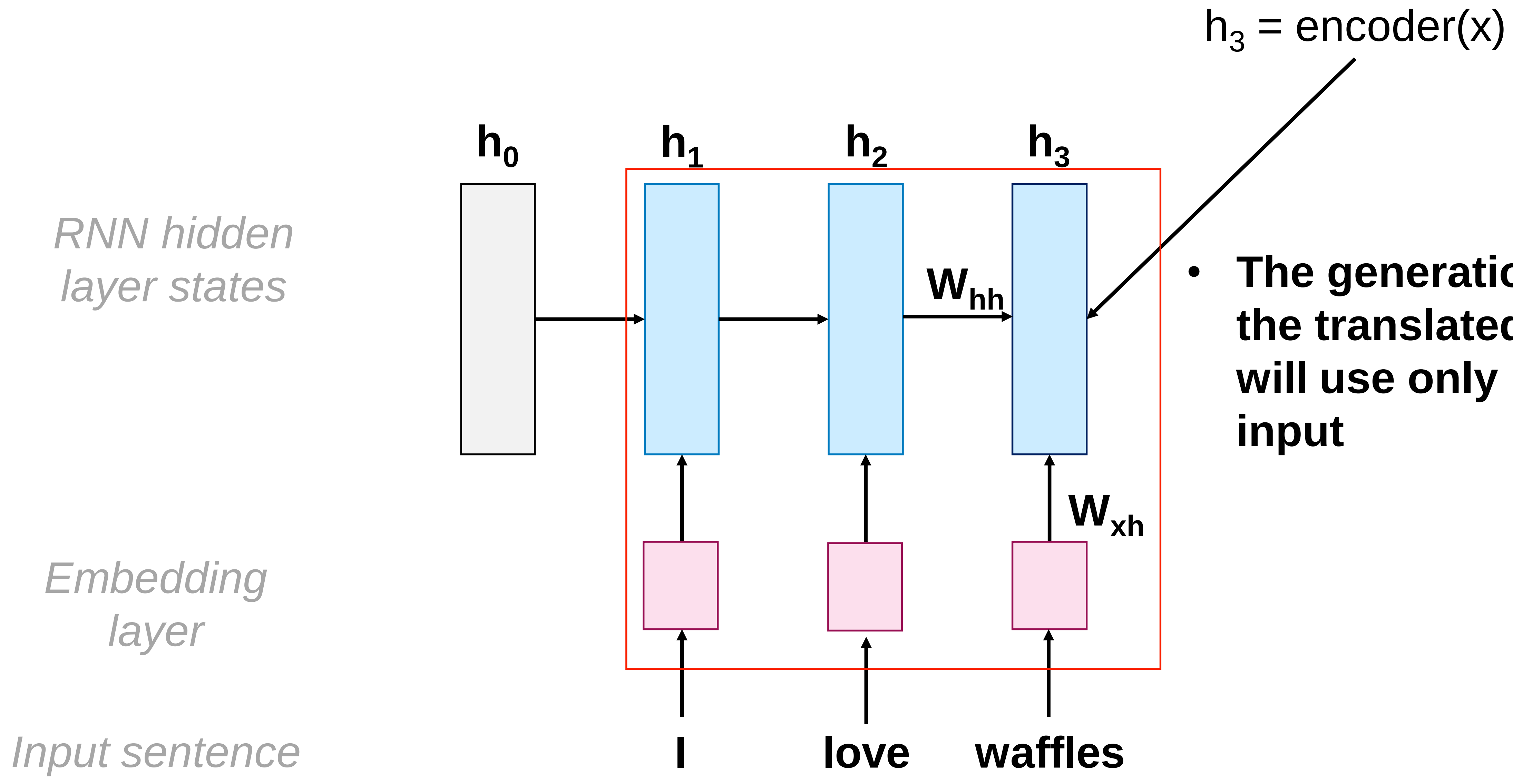
RNNs as encoders



RNNs as encoders

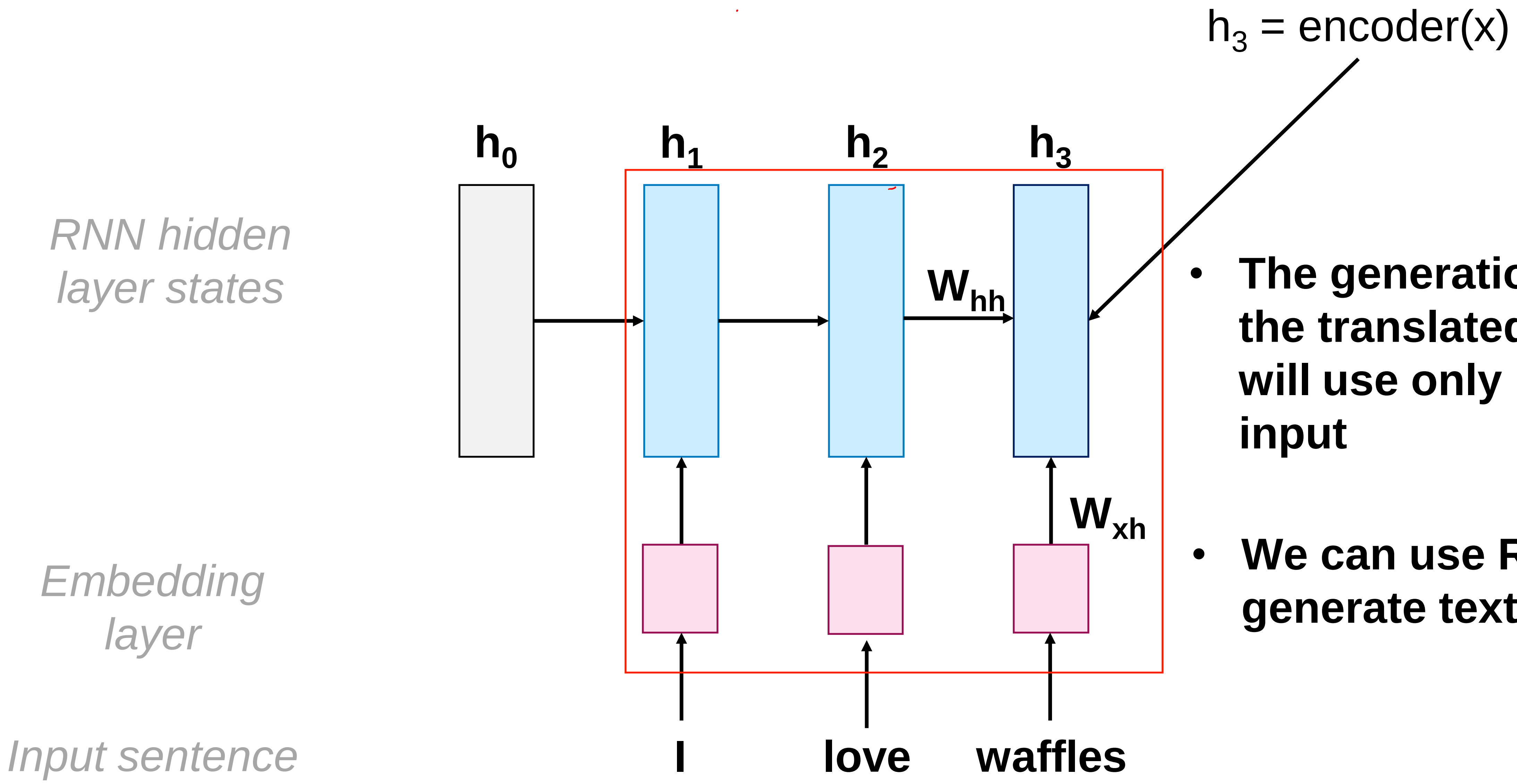


RNNs as encoders



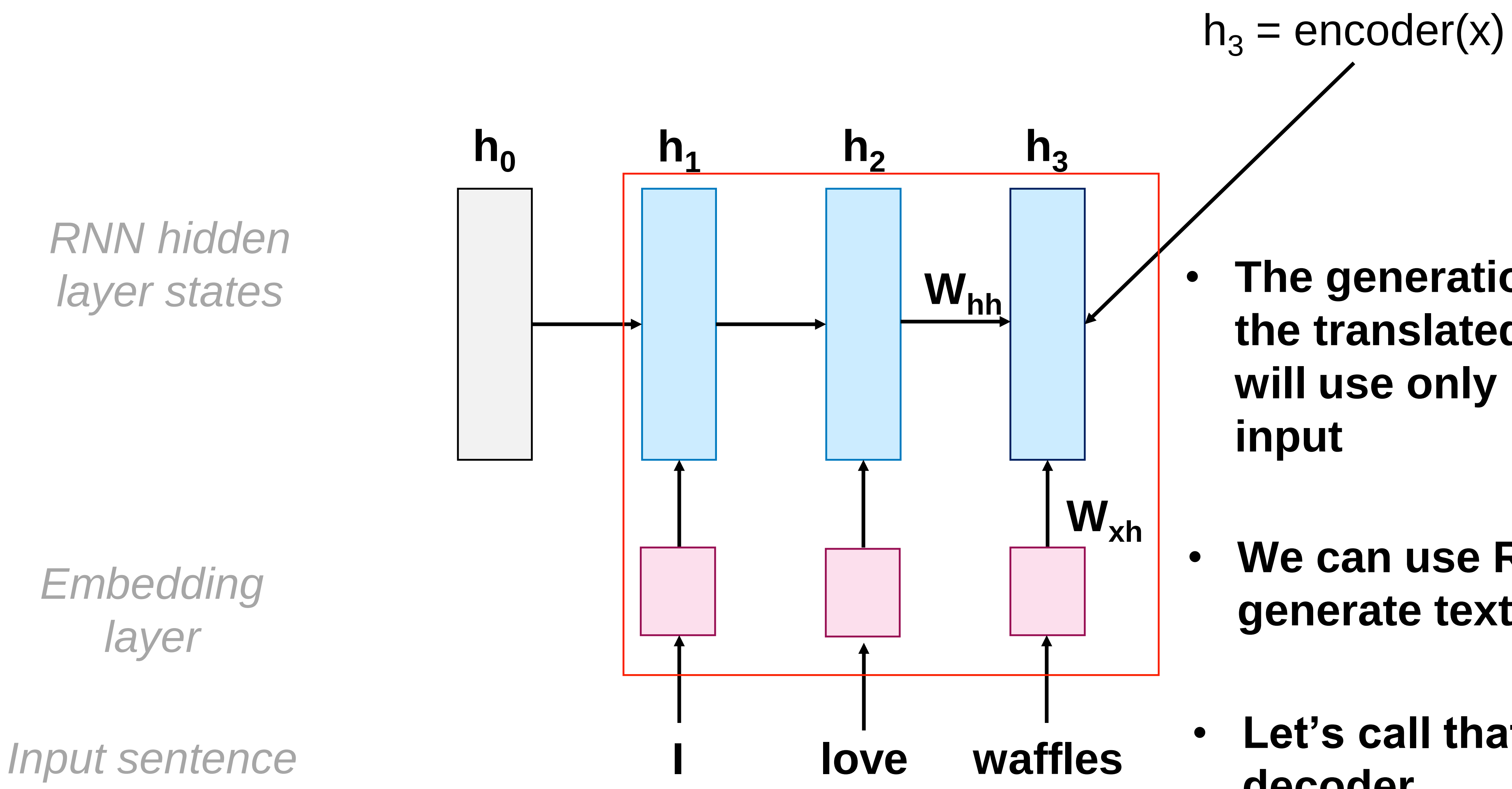
- **The generation of the translated text will use only h_3 as input**

RNNs as encoders



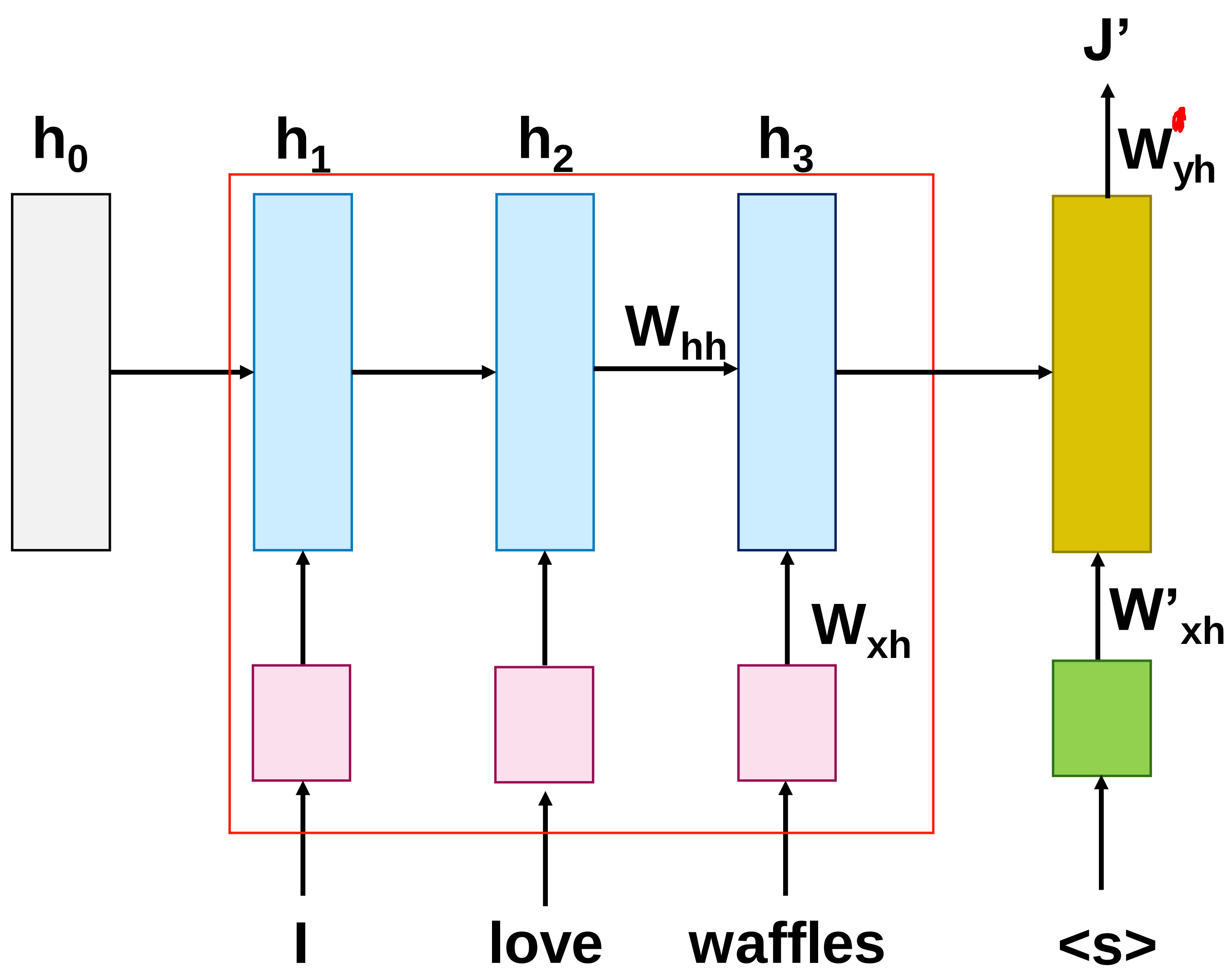
- **The generation of the translated text will use only h_3 as input**
- **We can use RNNs to generate text**

RNNs as encoders

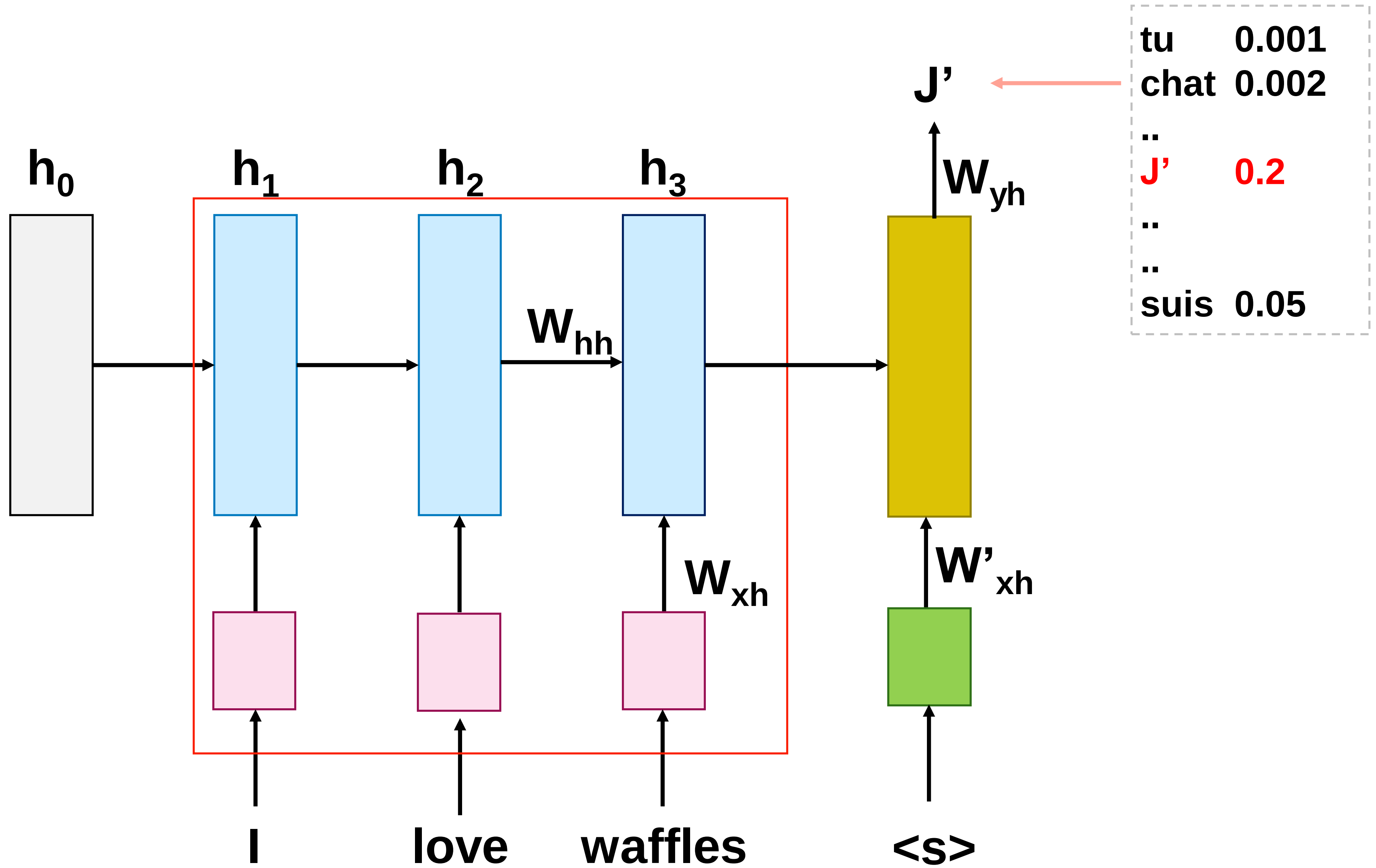


- **The generation of the translated text will use only h_3 as input**
- **We can use RNNs to generate text**
- **Let's call that a decoder**

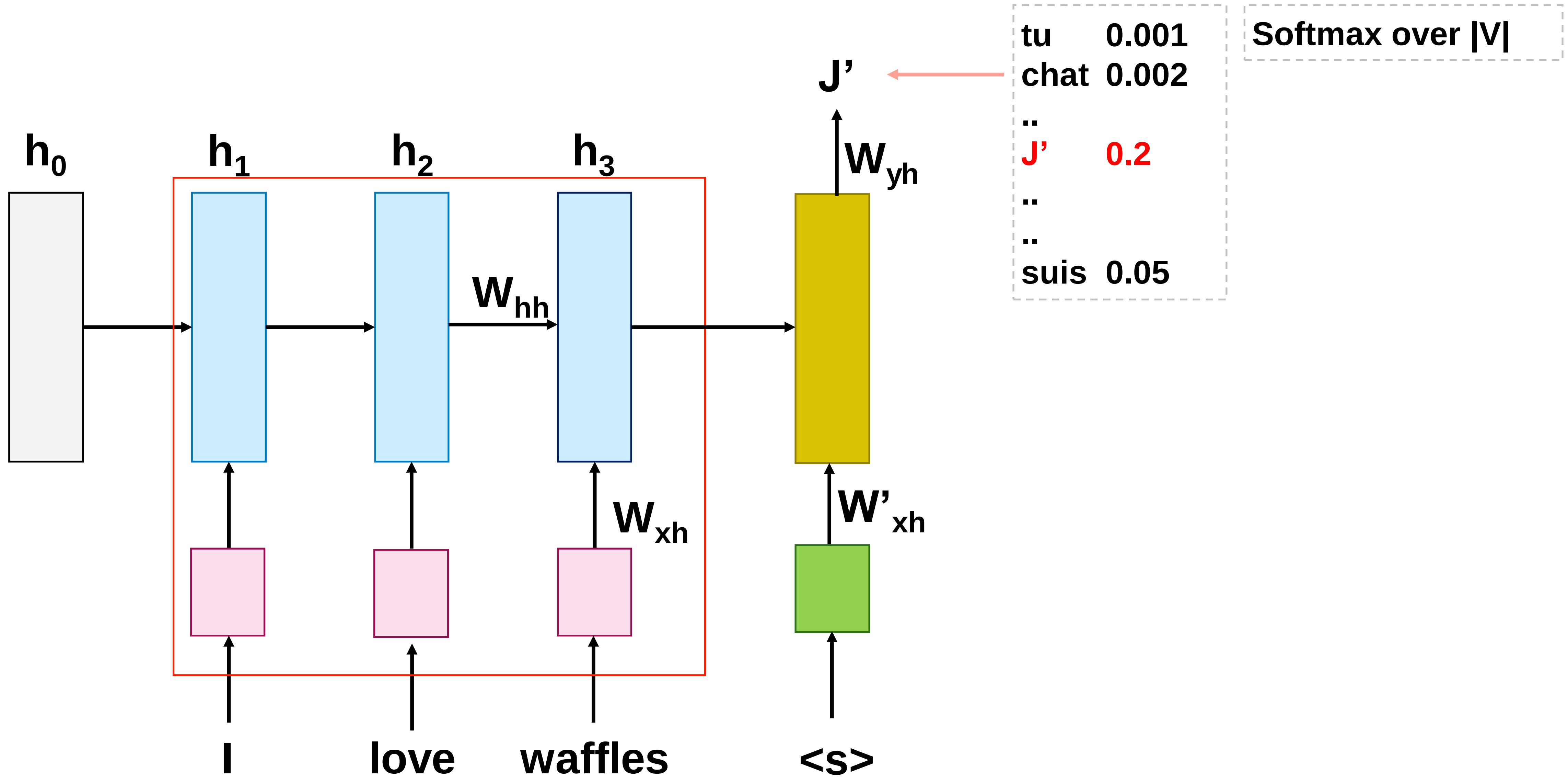
RNNs as decoders



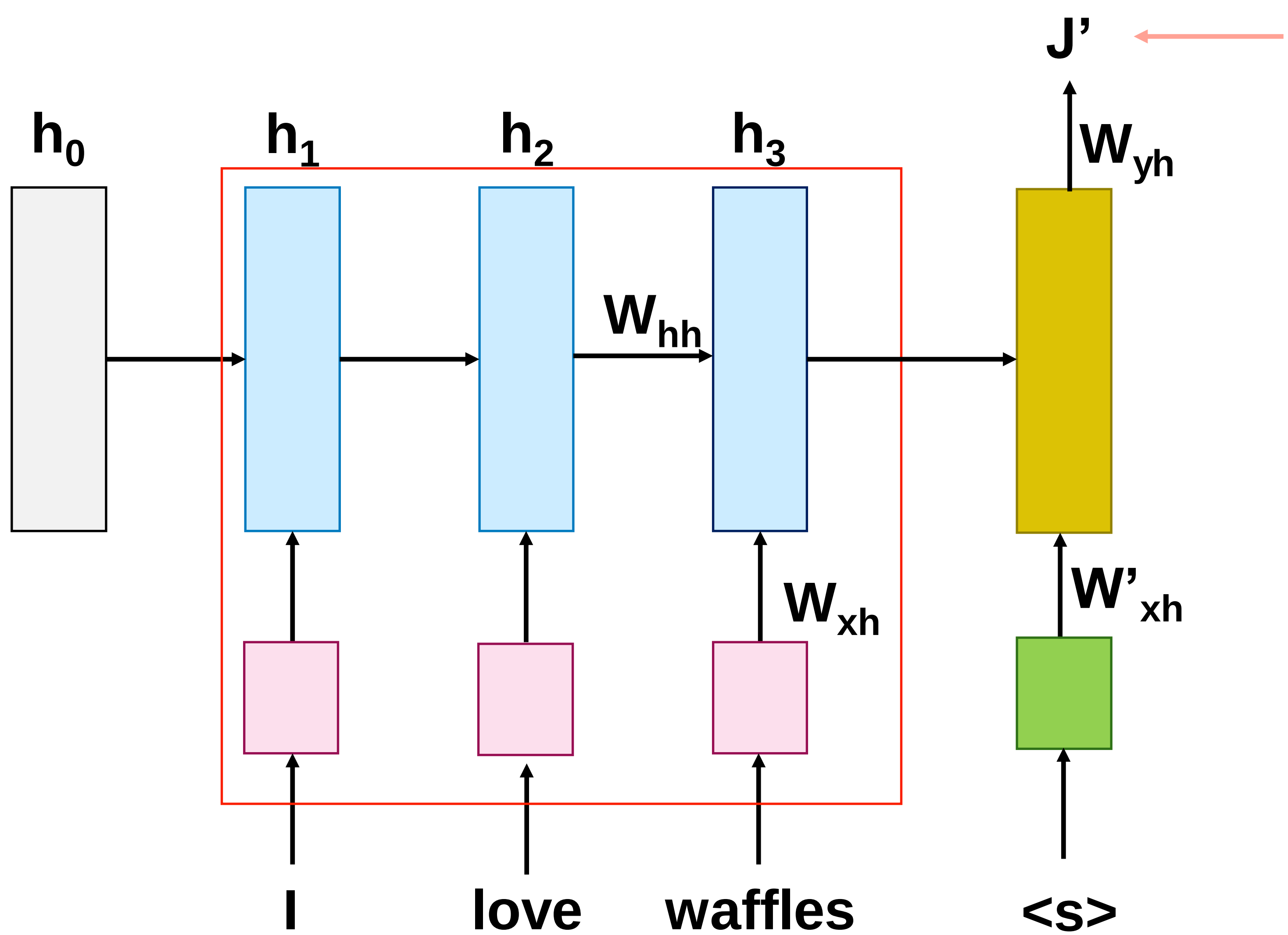
RNNs as decoders



RNNs as decoders



RNNs as decoders

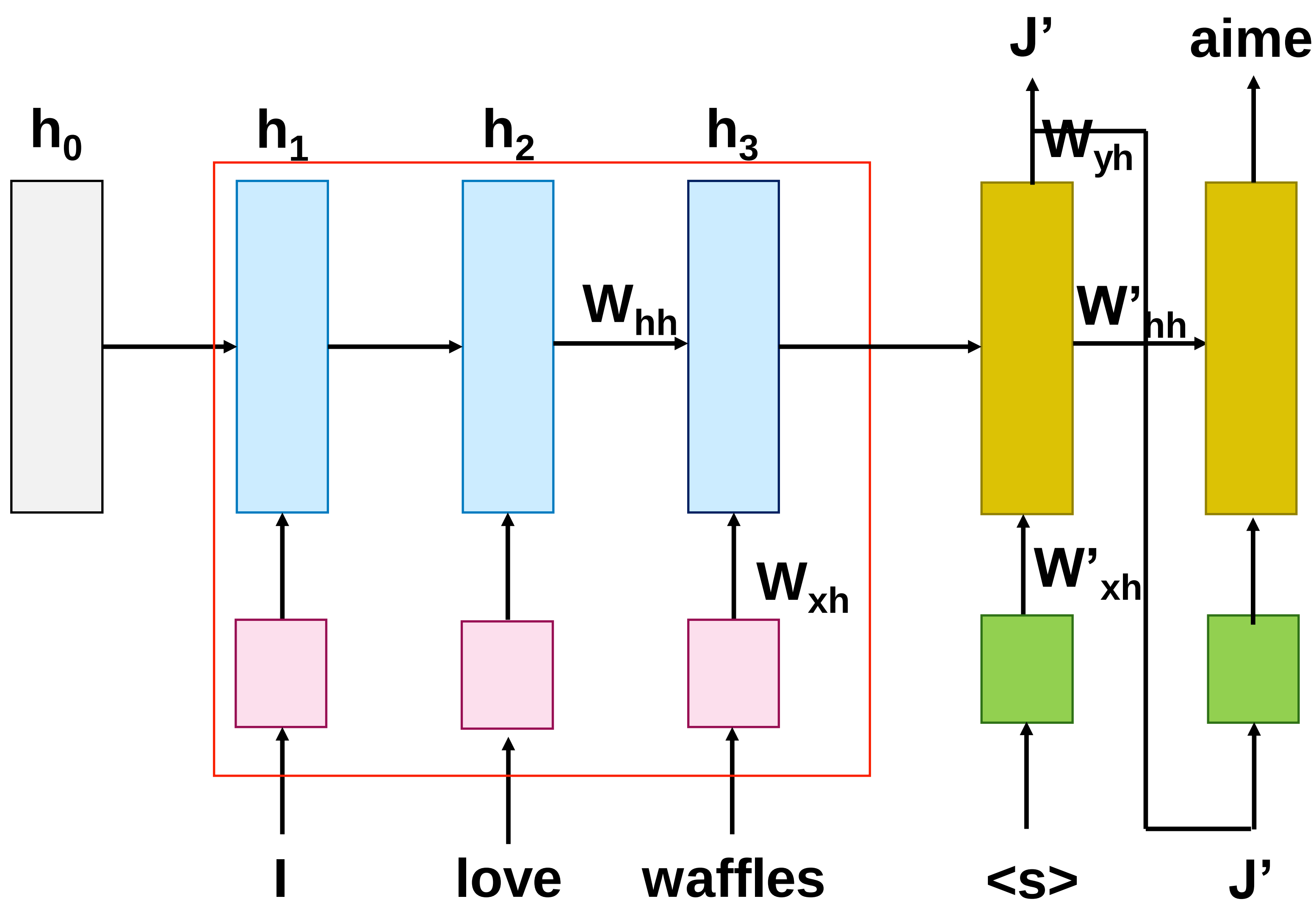


tu	0.001
chat	0.002
..	
J'	0.2
..	
..	
suis	0.05

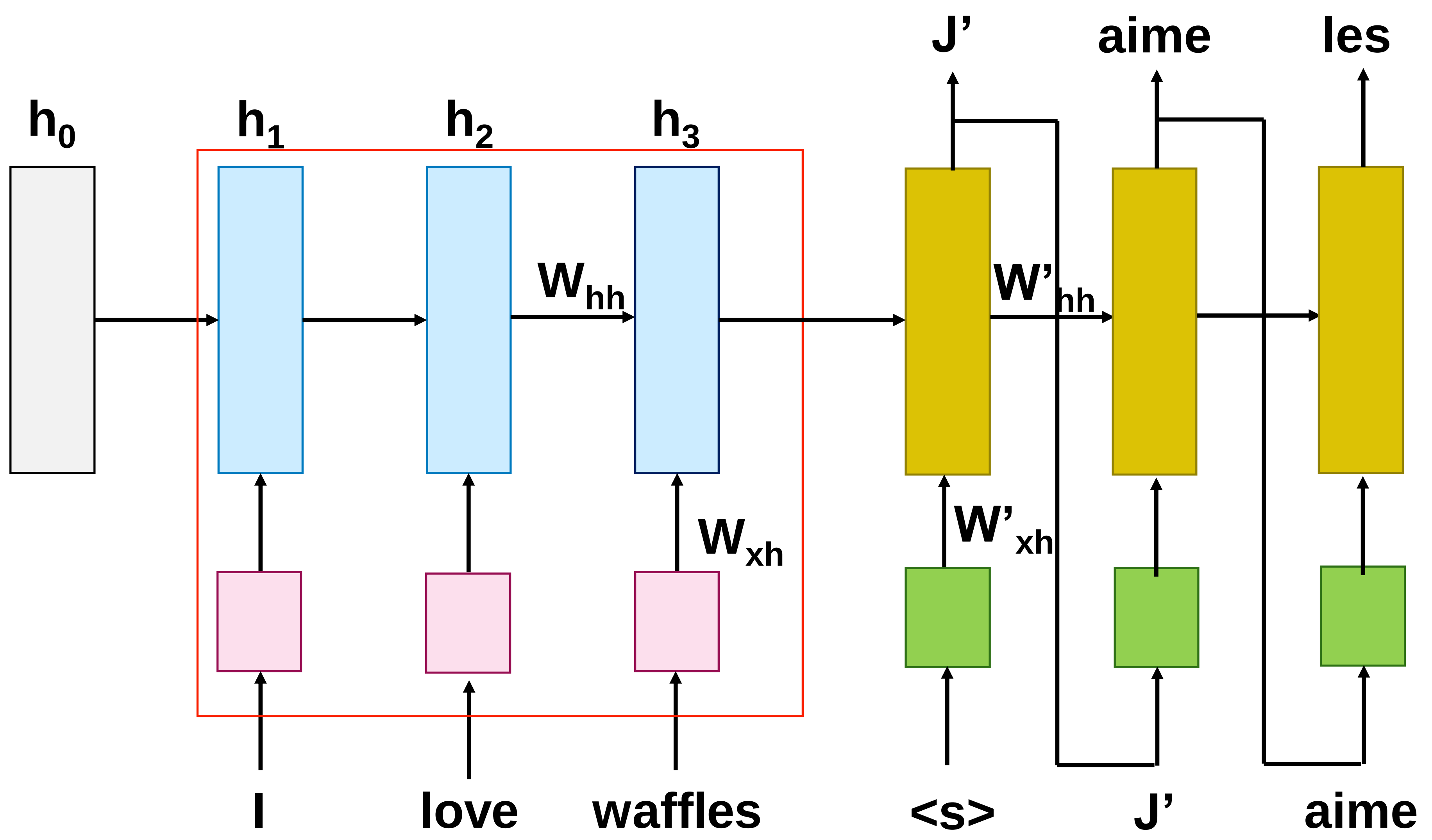
Softmax over |V|

N.B.: Crossentropy loss (-*log(prob)*) even though final system will be evaluated with BLEU.

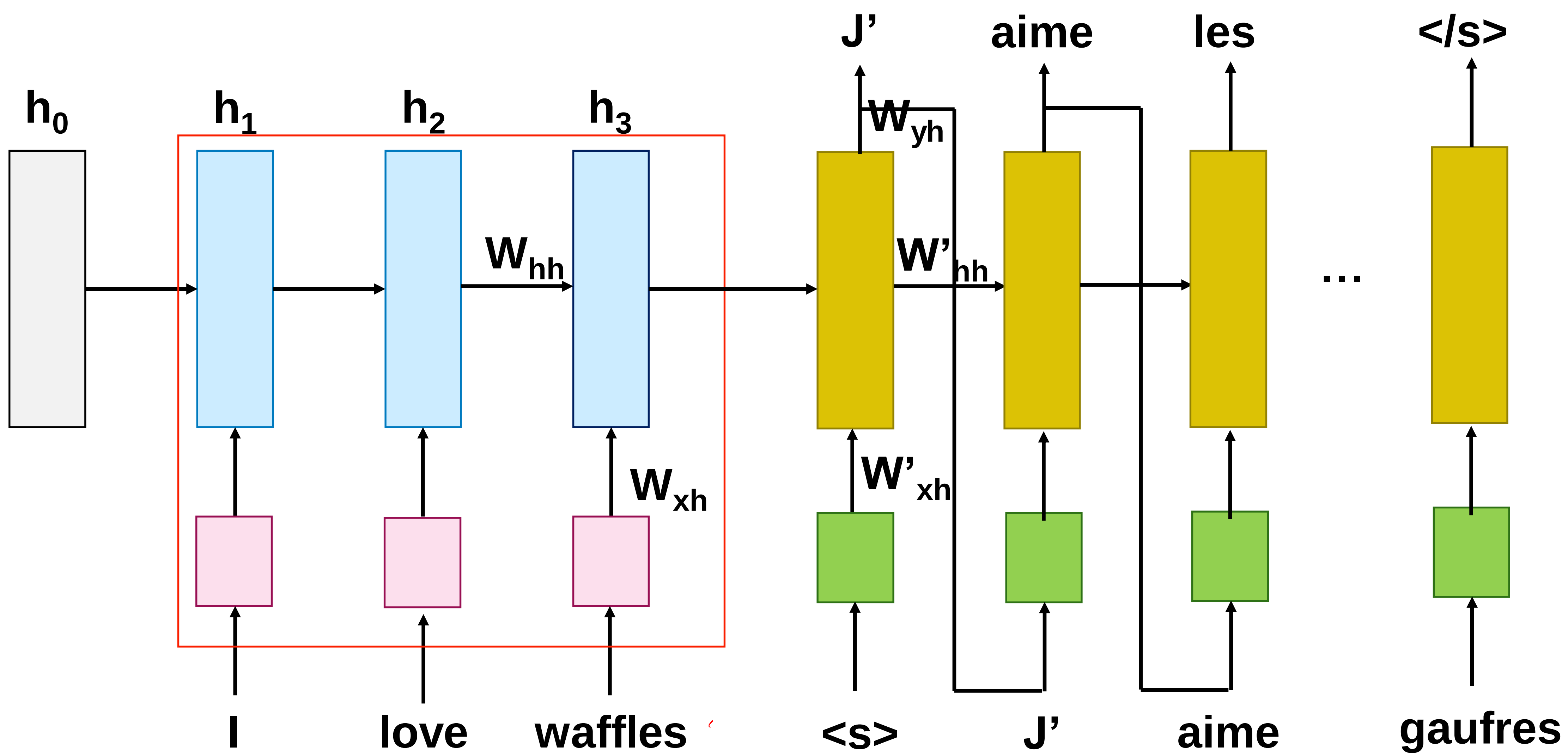
RNNs as decoders



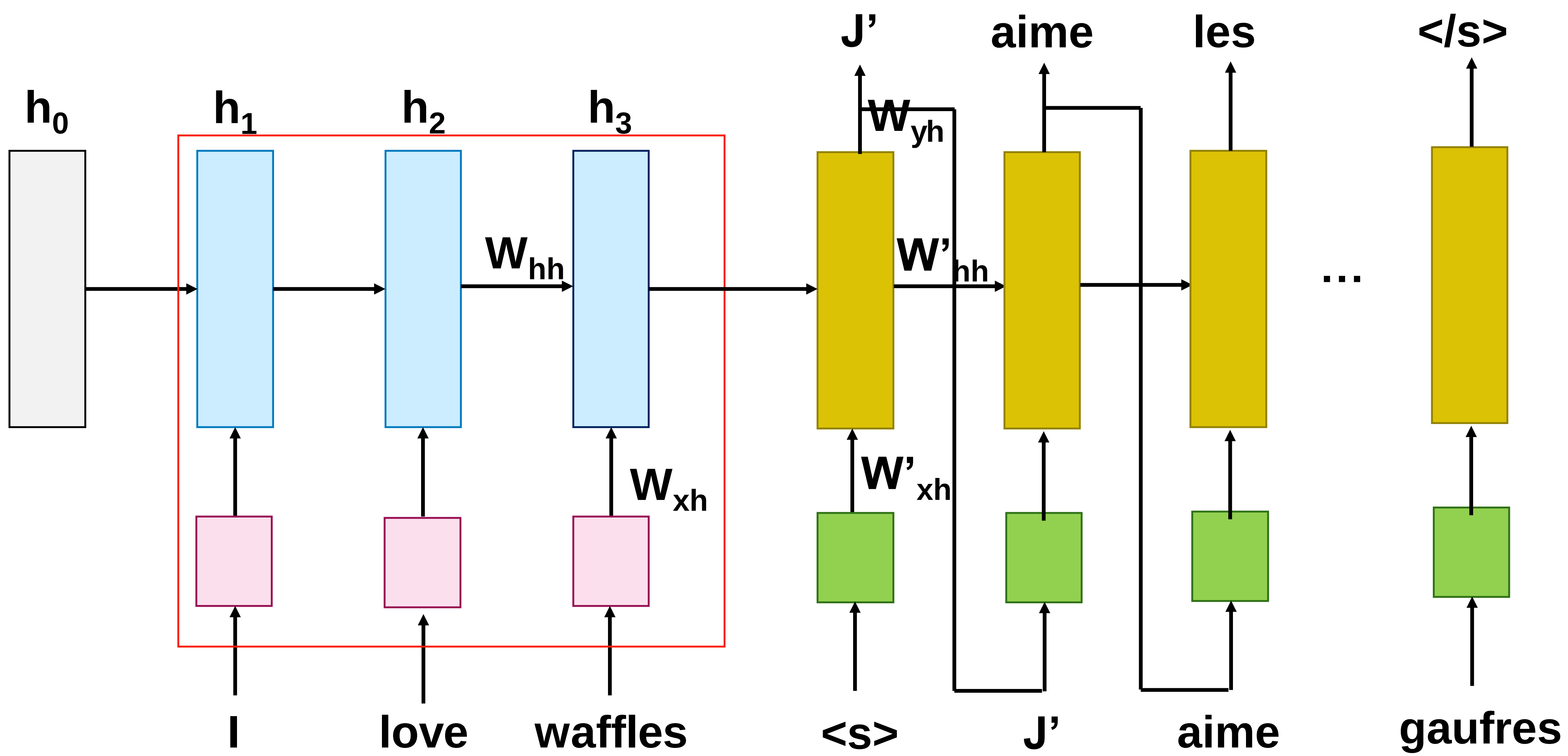
RNNs as decoders



RNNs as decoders



RNNs as decoders

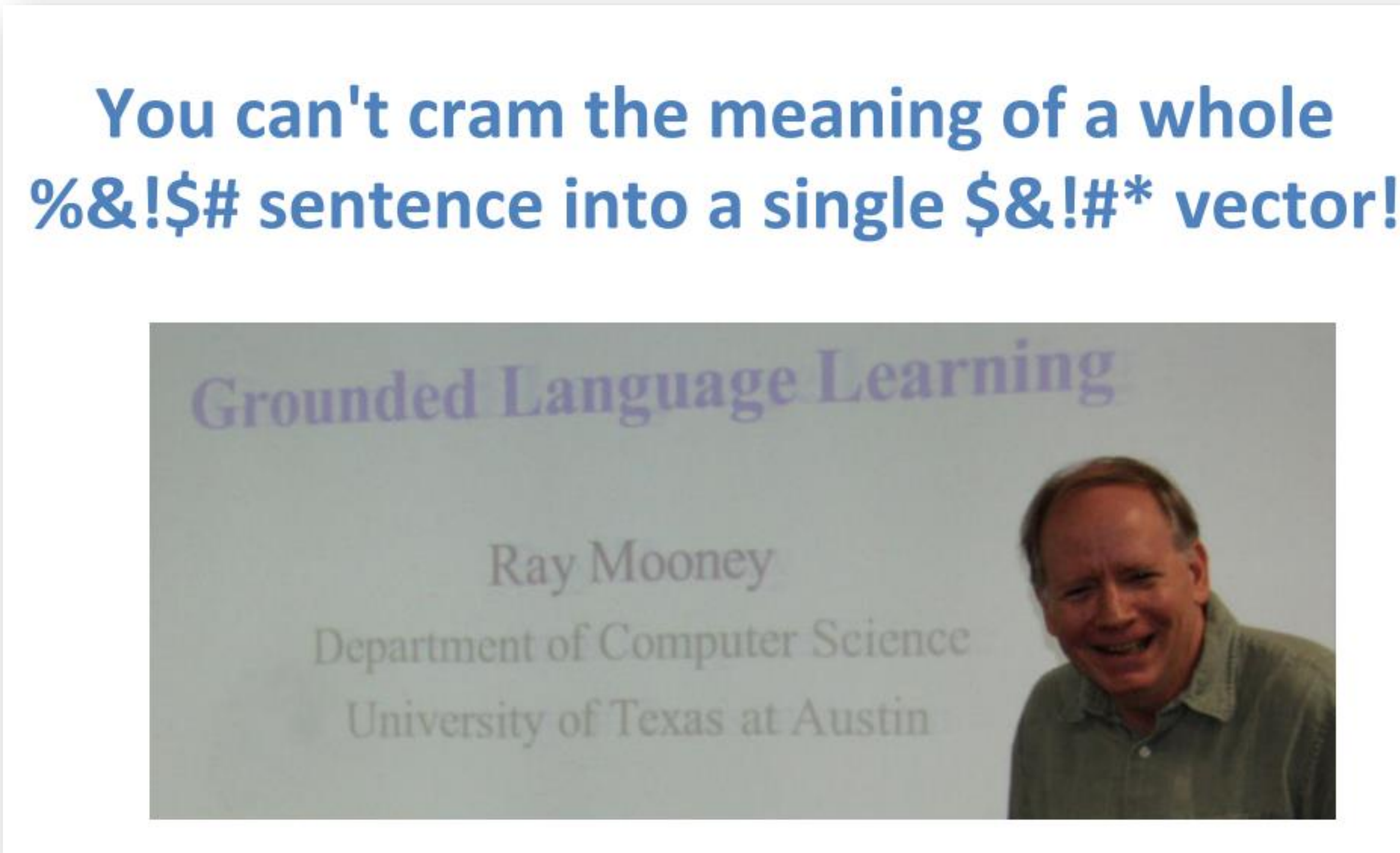


N.B.: Whether you're using a vanilla RNN or LSTM, the structure is the same. The only difference is to add the cell state to the picture.

Encoder decoder issues

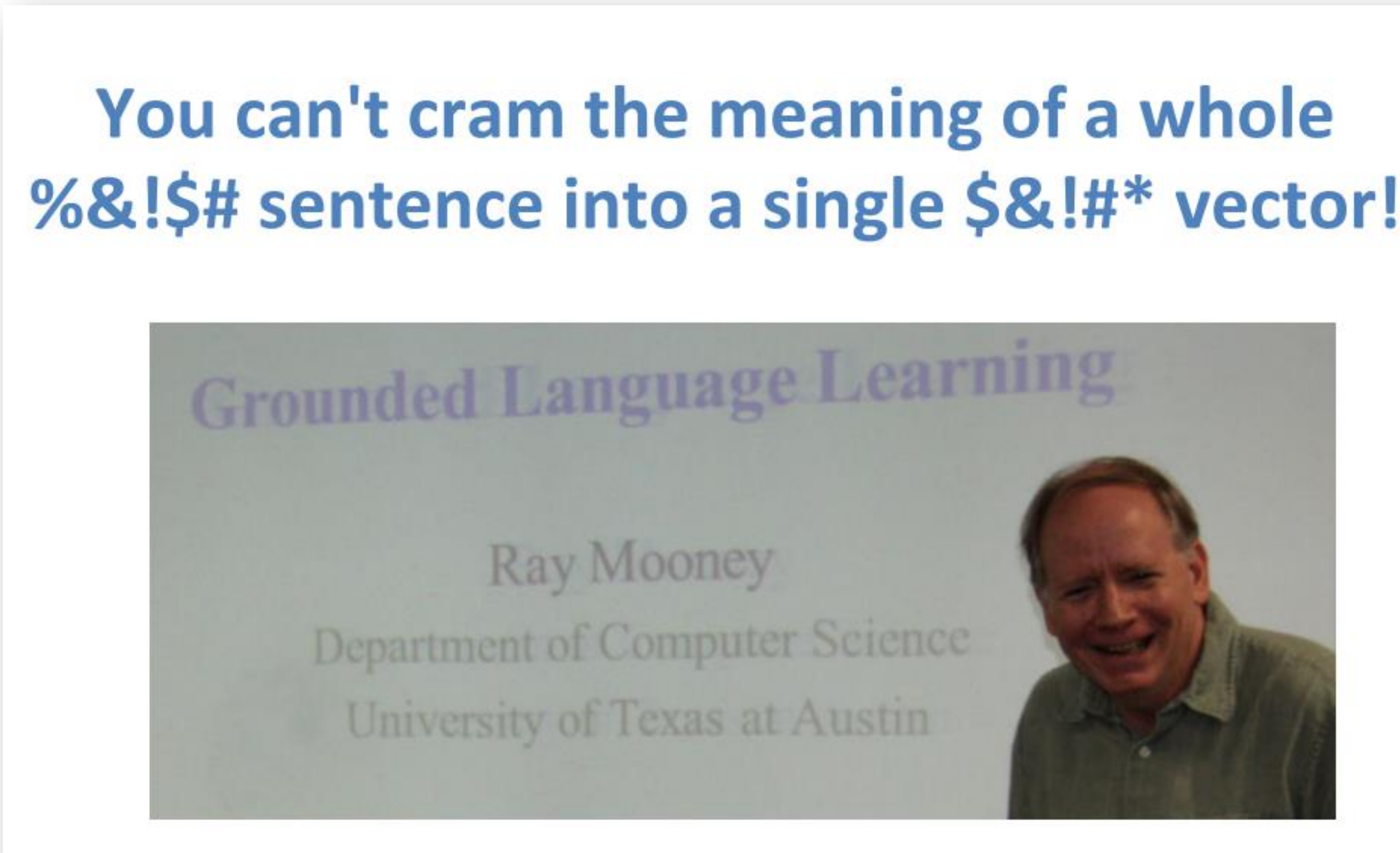
- Problems handling long sentences (Why?)
- Limited vocab size (Why?)
- .. And also:

Encoder decoder issues



https://cs224d.stanford.edu/lectures/CS224d-Lecture15.pdf

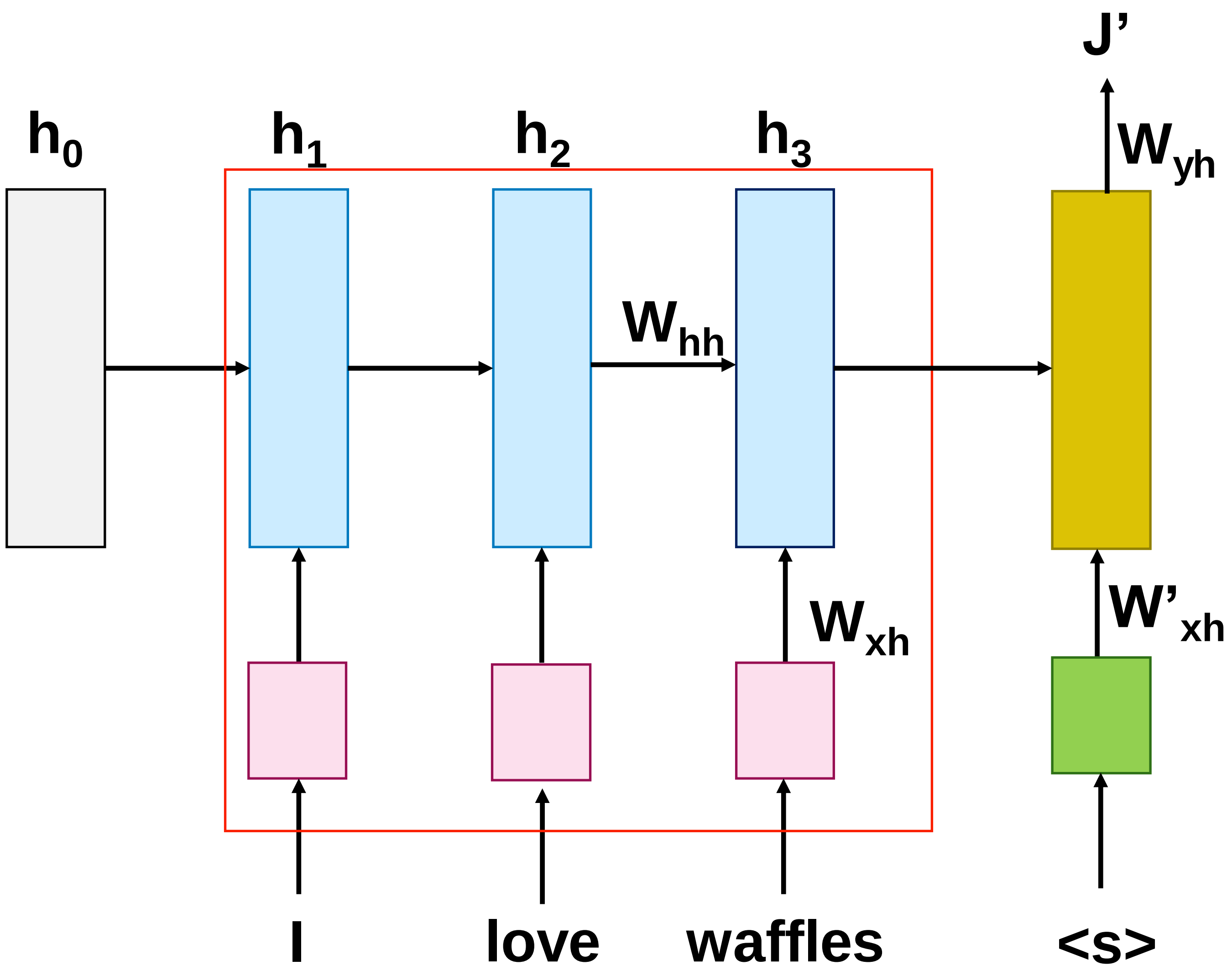
Encoder decoder issues



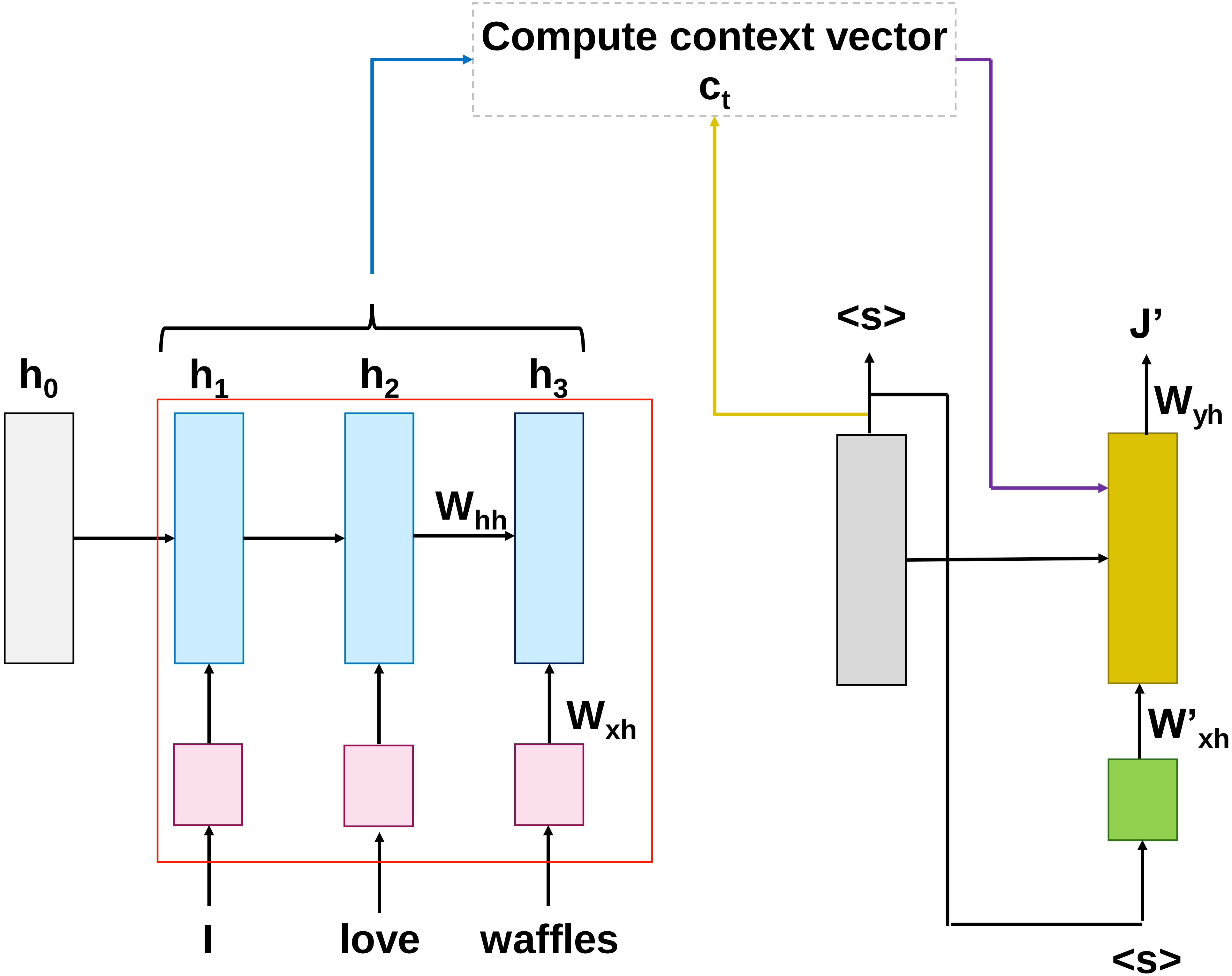
Attention to the rescue!

https://cs224d.stanford.edu/lectures/CS224d-Lecture15.pdf

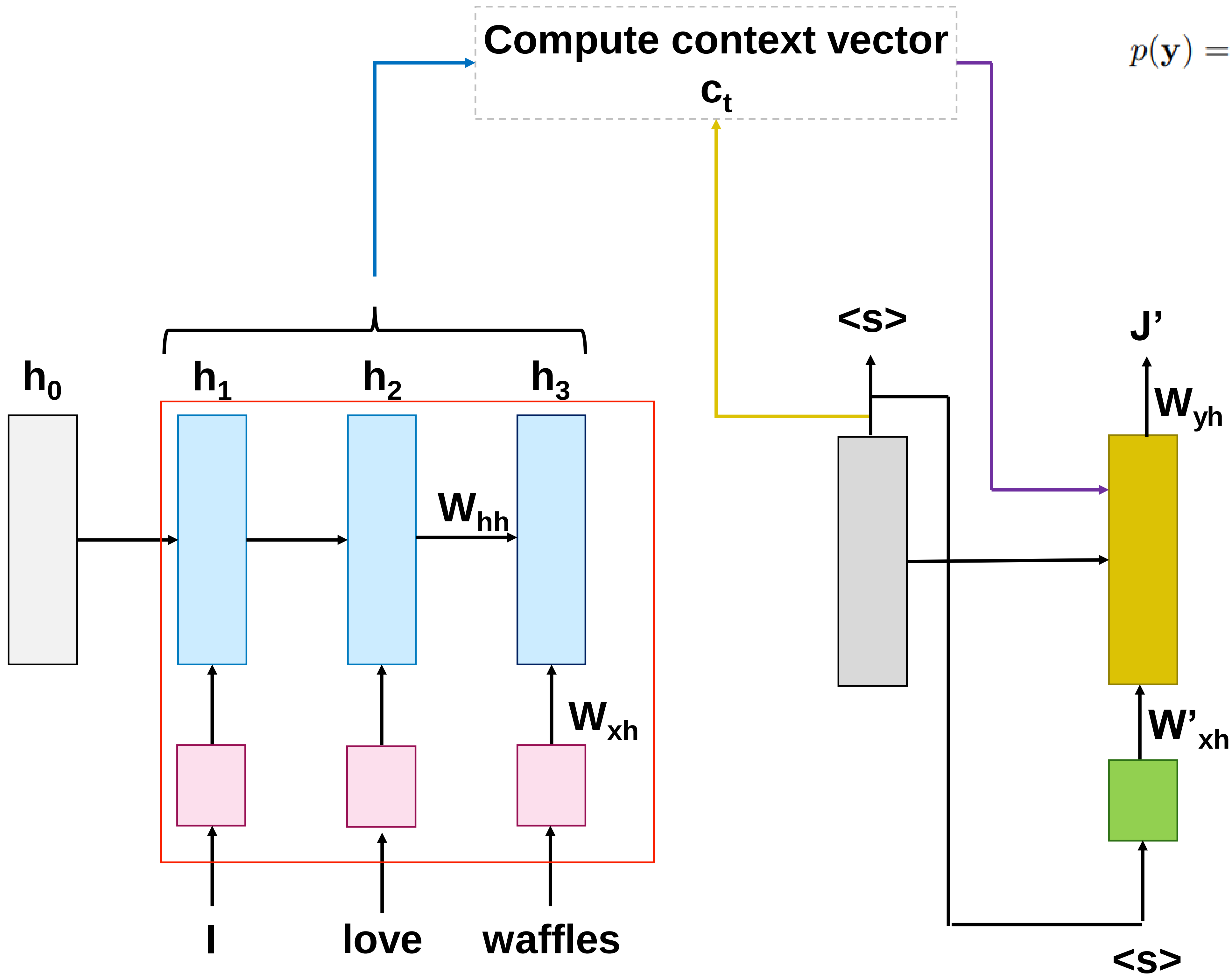
Attention mechanism



Attention mechanism



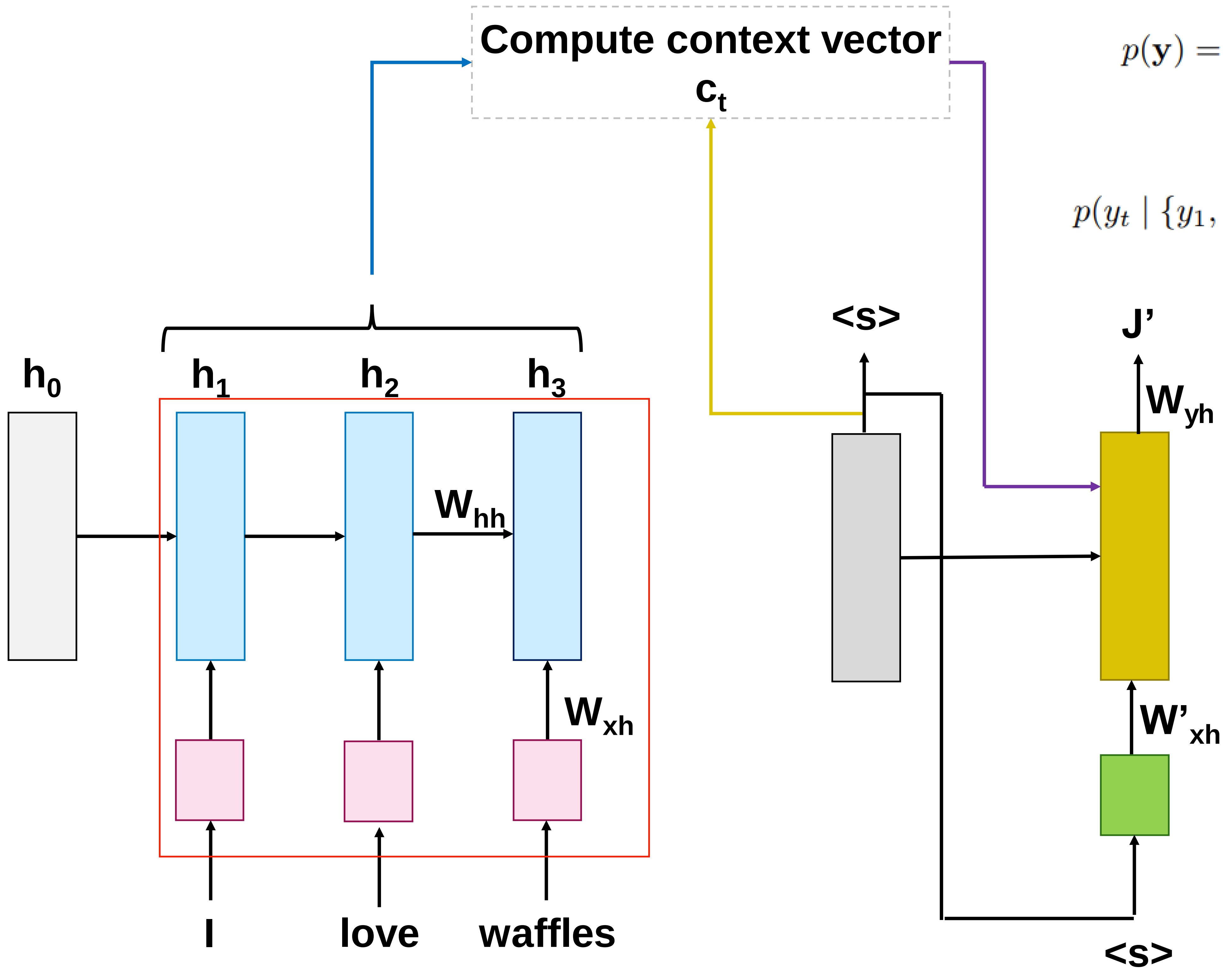
Attention mechanism



$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism

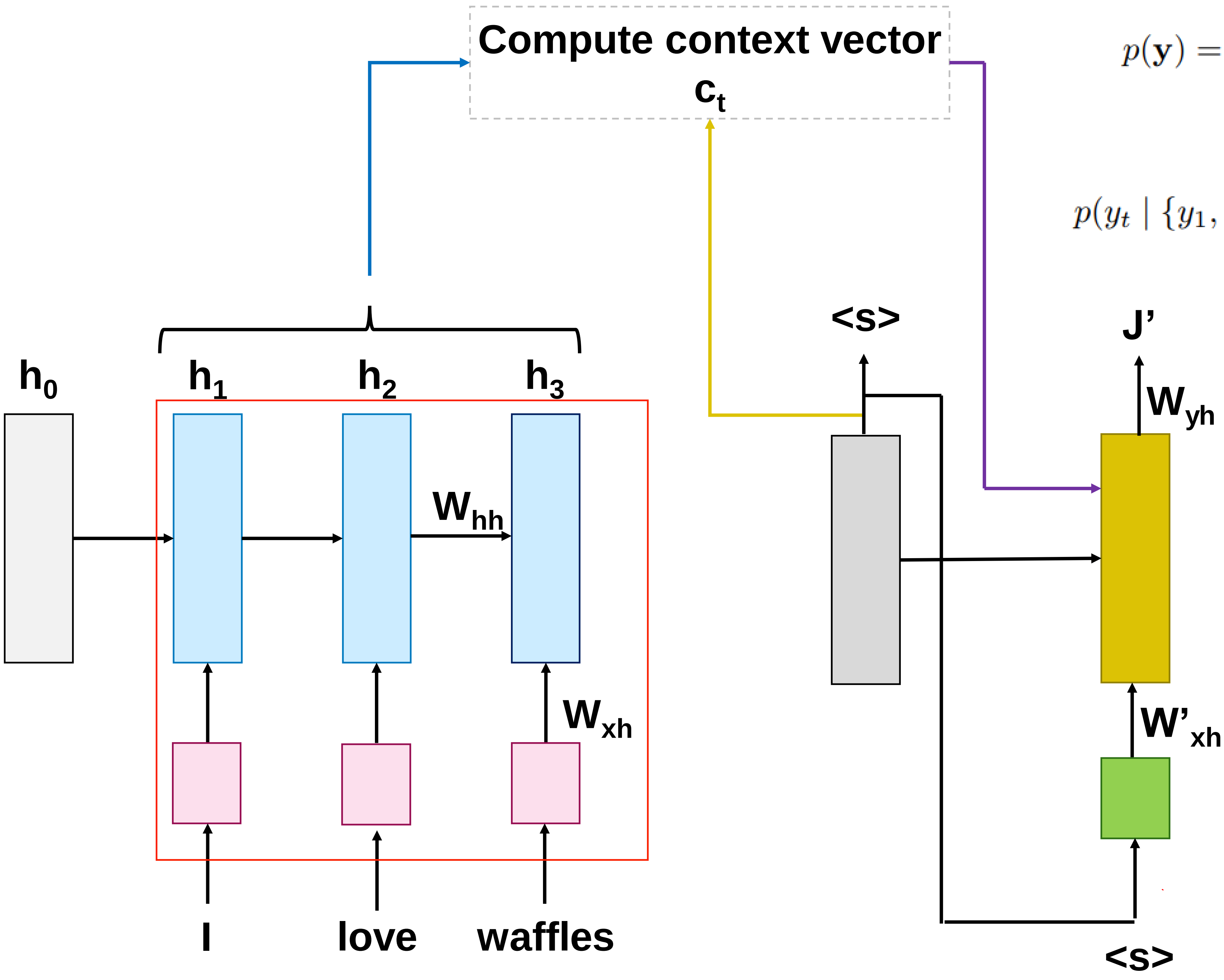


$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism



$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

$$\downarrow$$

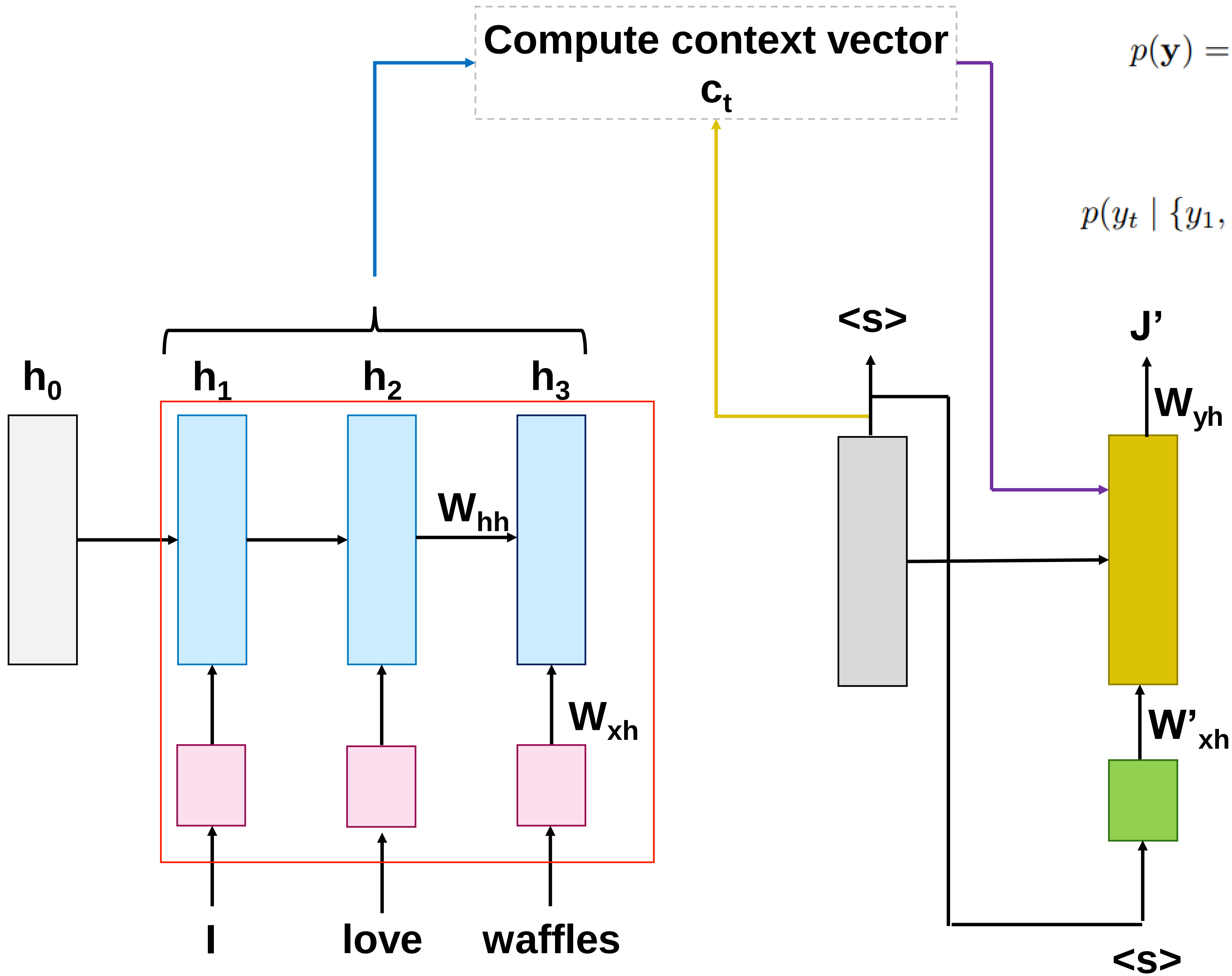
$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

$$\downarrow$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism



$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

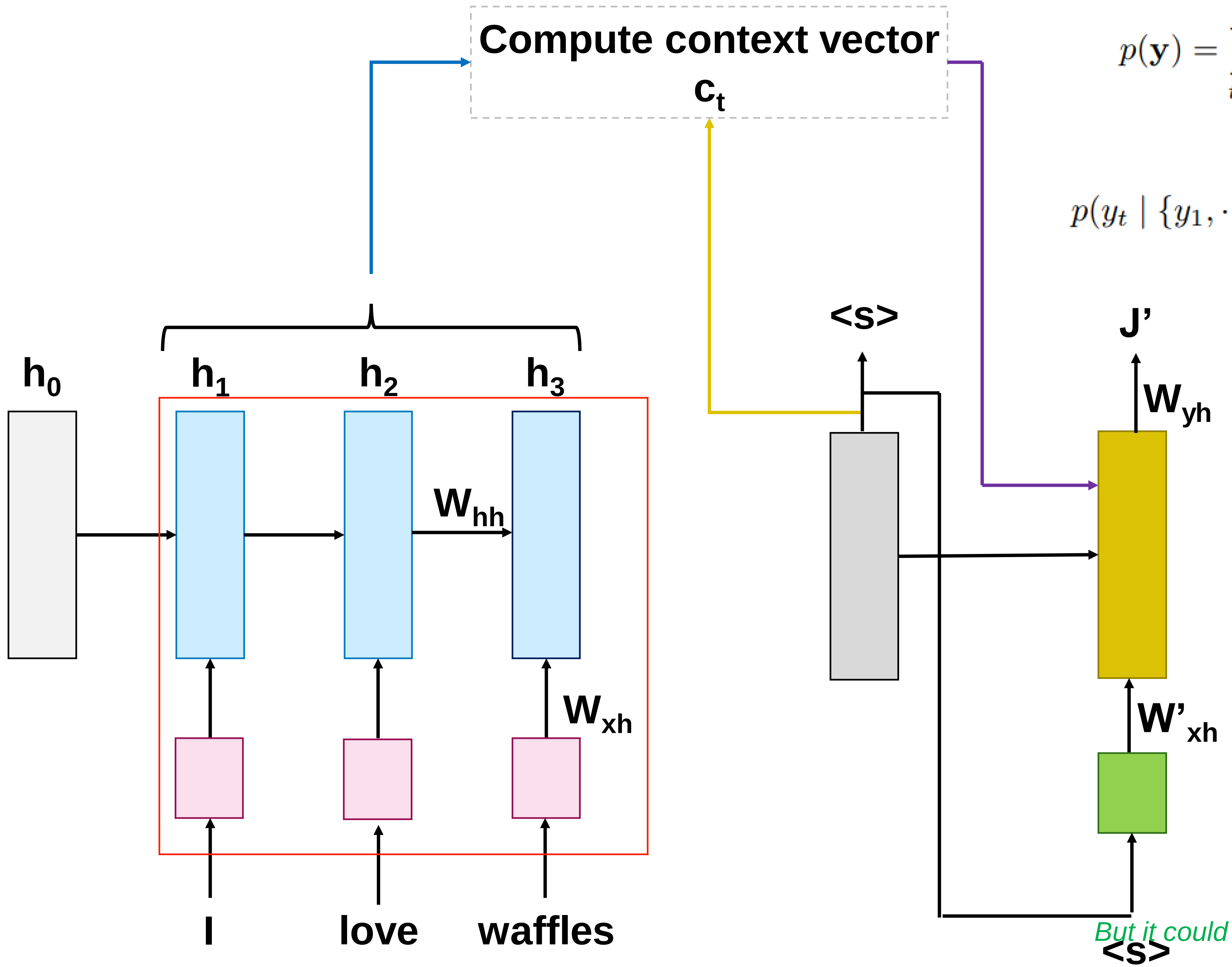
$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

$$s_i = f(s_{i-1}, y_{i-1}, \boxed{c_i}).$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism



$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

$$\downarrow$$

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

$$\downarrow$$

$$s_i = f(s_{i-1}, y_{i-1}, \boxed{c_i}).$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\downarrow$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$\downarrow$$

$$e_{ij} = \boxed{a(s_{i-1}, h_j)}$$

Feed forward model
1 hidden layer
But it could be anything that measures similarity
Dot product, cosine, etc

https://arxiv.org/pdf/1409.0473.pdf

MT Evaluation

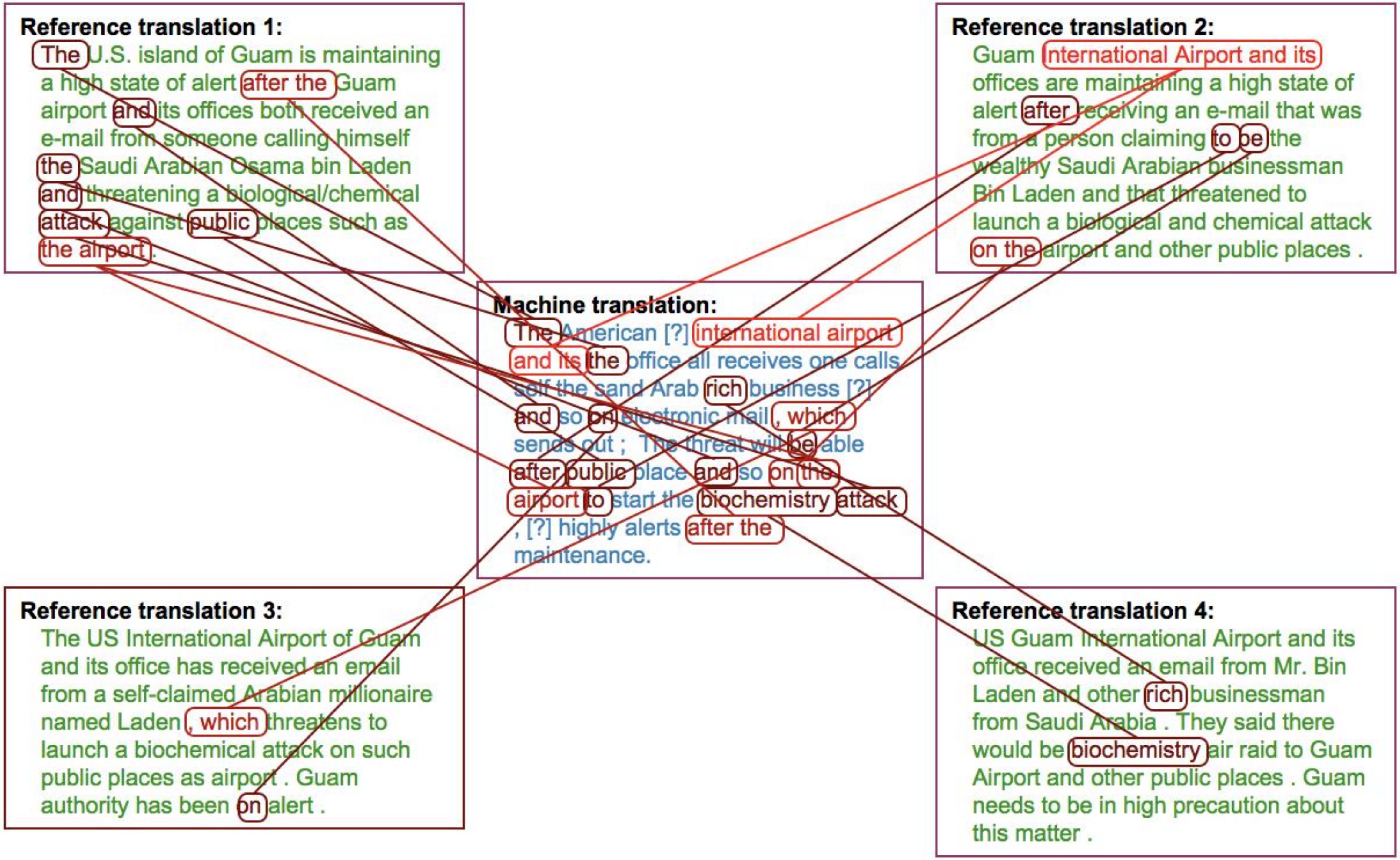
- MT Evaluation is notoriously difficult!
- No single correct output (typically use multiple reference translations).
- Need to evaluate faithfulness and fluency, but both are subjective.
- Dumb machines vs. slow humans.
- Wide range of different automatic metrics (BLEU, NIST, TER, METEOR)

BLEU Metric

- **B**i**L**ingual **E**valuation **U**nderstudy
- Modified n-gram precision with length penalty. Recall is ignored.
- Quick, inexpensive, and language independent.
- Correlates highly with human evaluations.
- But: Bias against synonyms and inflectional variations. Penalizes variations in word-order between languages in different families.

(Papineni, et al. 2002. "BLEU: A method for automatic evaluation of machine translation.")
https://www.aclweb.org/anthology/P02-1040.pdf

Multiple Reference Translations



Slide from Bonnie Dorr

Computing BLEU

- Test sentence:
  Colorless green ideas sleep furiously

- Reference translations:
  all dull jade ideas sleep irately
  drab emerald concepts sleep furiously
  colorless immature thoughts nap angrily

Computing BLEU

- Test sentence:
  Colorless *green* ***ideas*** ***sleep*** ***furiously***

- Reference translations:
  all dull jade ***ideas*** *sleep irately*
  drab emerald concepts ***sleep*** ***furiously***
  colorless *immature thoughts nap angrily*

Unigram precision: 4/5

Computing BLEU

- Test sentence:
  Colorless green ***ideas sleep*** *furiously*
  sleep furiously
- Reference translations:
  all dull jade ***ideas sleep*** *irately*
  drab emerald concepts ***sleep furiously***
  colorless immature thoughts nap angrily

Unigram precision: 4/5

Bigram precision: 2/4

Computing BLEU

- Test sentence:
  *Colorless green **ideas sleep** furiously*
  sleep furiously
- Reference translations:
  *all dull jade **ideas sleep** irately*
  *drab emerald concepts **sleep furiously***
  colorless immature thoughts nap angrily

Unigram precision: 4/5 =0.8

Bigram precision: 2/4 =0.5

BLEU score = $(a_1 \times a_2 \times_{\dots} \times a_n)^{1/n}$

$= (0.8 \times 0.5)^{1/2} = 0.6325 \rightarrow$ **63.25**

BLEU: Brevity Penalty

- BLEU is precision based. Dropped words are not penalized.

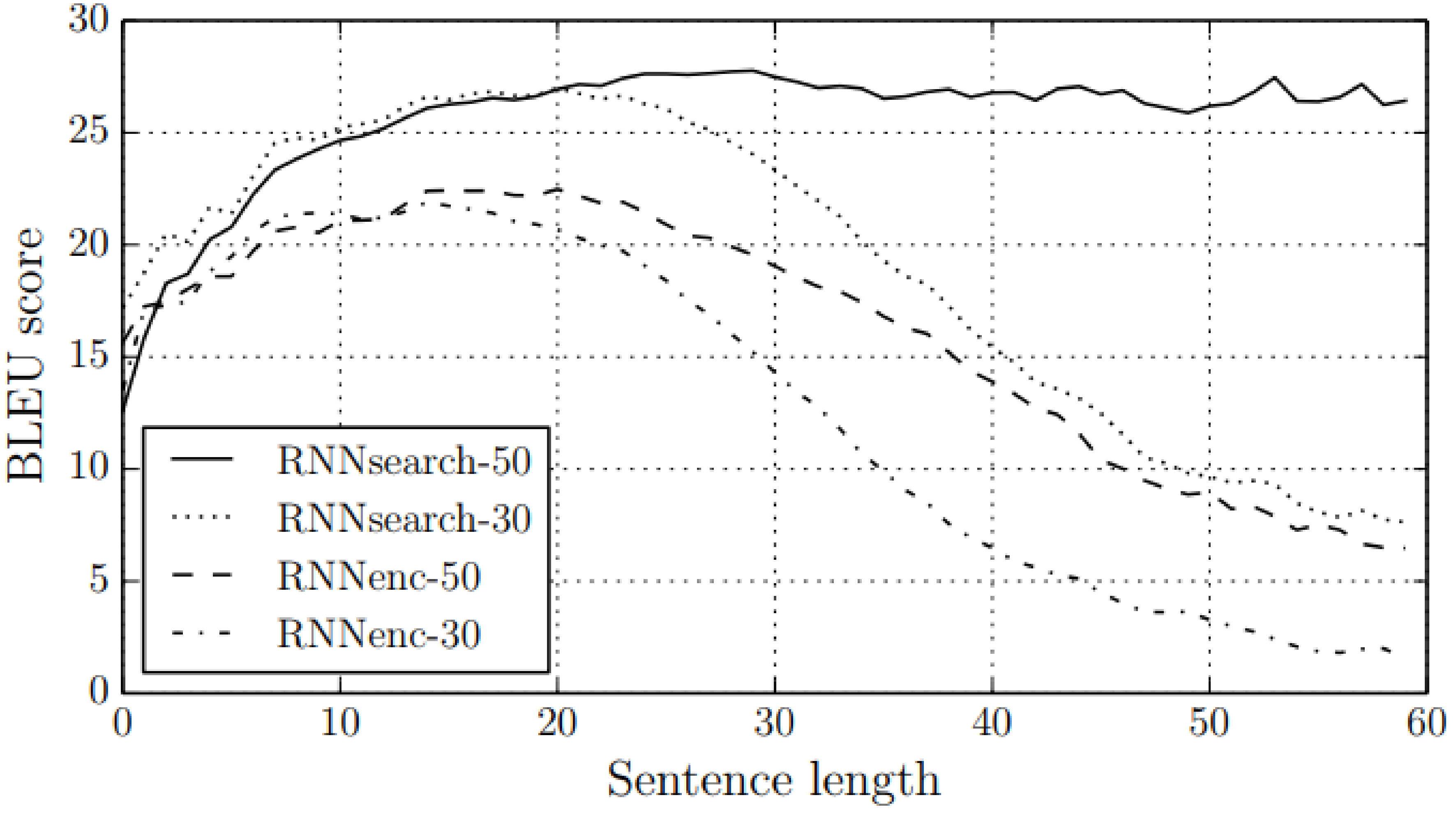
- Instead, a **brevity penalty** is used for translations that are shorter than the reference translations.

- Let c be the length of the candidate translation and r be the length of the reference translation that has the closest length.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

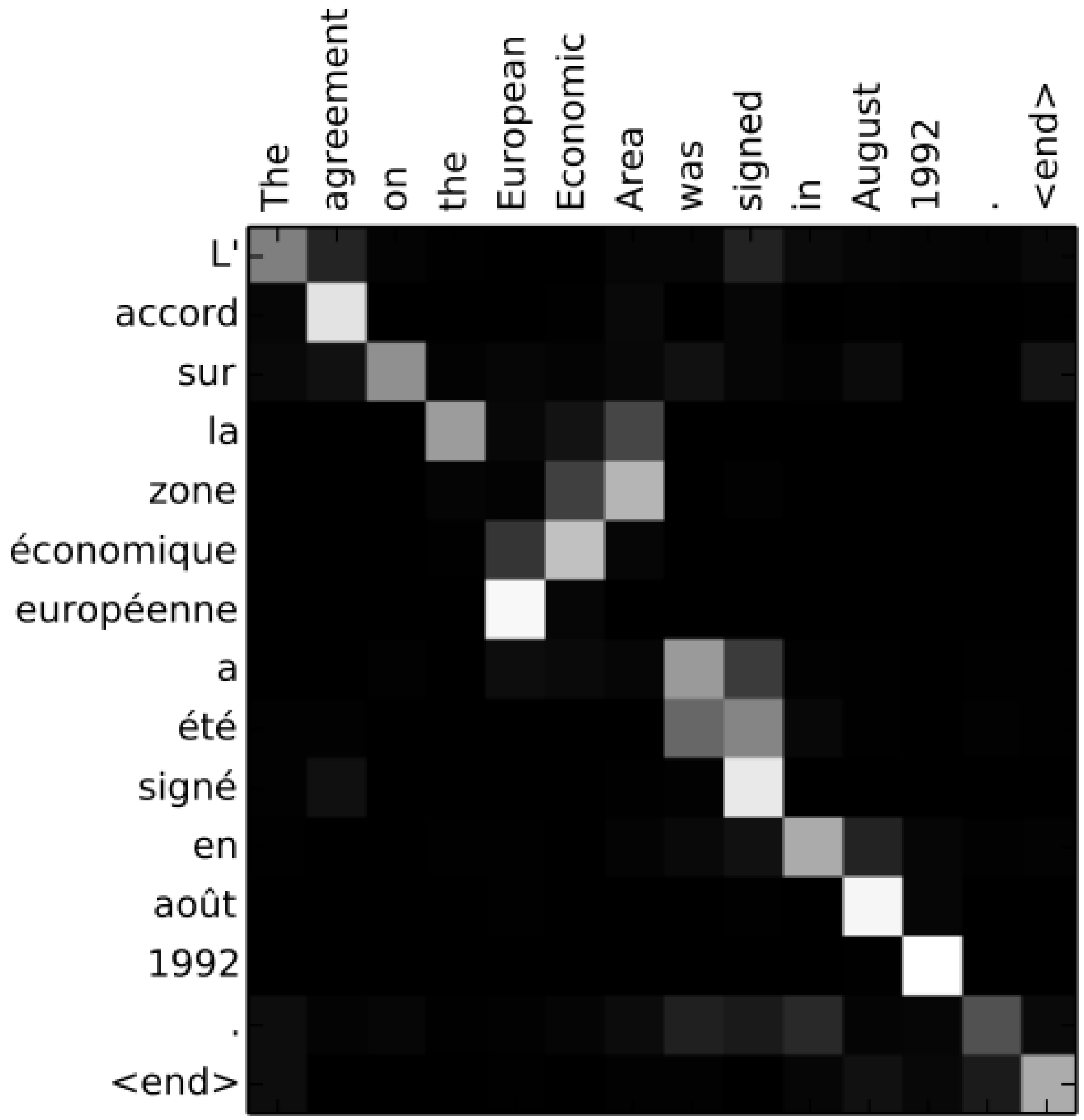
$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} \frac{1}{n} \log\ \text{precision}_n\right)$$

Attention mechanism



https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism



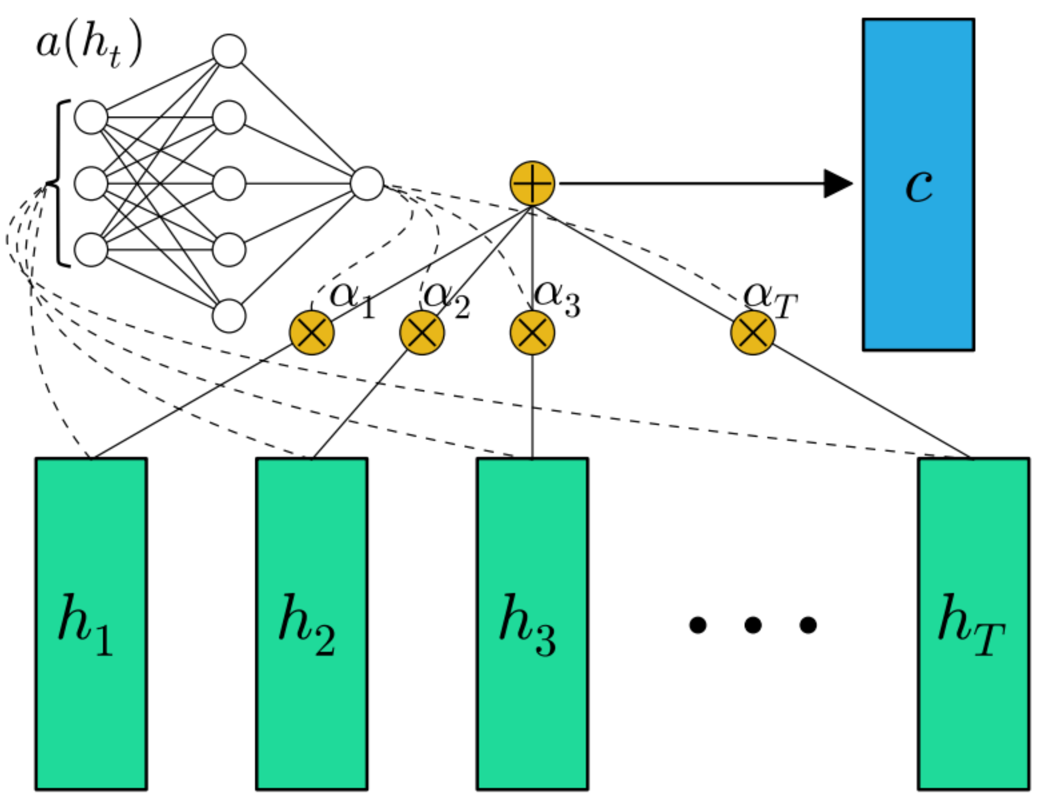
https://arxiv.org/pdf/1409.0473.pdf

Attention mechanism

What if we have a classification task? What is different?

Attention mechanism

What if we have a classification task? What is different?



Self - attention

- Why do we need it?
  - Despite the success of Gated RNNs with attention, you still have a limitation on speed since we need to process a sentence sequentially.
  - Self – attention is a way to get the same results without this constraint.
  - Let’s talk a look through an example

Self – attention ABAE

An Unsupervised Neural Attention Model for Aspect Extraction

Ruidan He[†‡], **Wee Sun Lee**[†], **Hwee Tou Ng**[†], and **Daniel Dahlmeier**[‡]
[†]Department of Computer Science, National University of Singapore
[‡]SAP Innovation Center Singapore
[†]{ruidanhe,leews,nght}@comp.nus.edu.sg
[‡]d.dahlmeier@sap.com

Abstract

Aspect extraction is an important and challenging task in aspect-based sentiment analysis. Existing works tend to apply variants of topic models on this task. While fairly successful, these methods usually do not produce highly coherent aspects. In this paper, we present a novel neural approach with the aim of discovering coherent aspects. The model improves coherence by exploiting the distribution of word co-occurrences through the use of neural word embeddings. Unlike

aspect (e.g., cluster *"beef", "pork", "pasta"*, and *"tomato"* into one aspect *food*).

Previous works for aspect extraction can be categorized into three approaches: rule-based, supervised, and unsupervised. Rule-based methods usually do not group extracted aspect terms into categories. Supervised learning requires data annotation and suffers from domain adaptation problems. Unsupervised methods are adopted to avoid reliance on labeled data needed for supervised learning.

In recent years, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and its variants (Titov and McDonald, 2008; Brody and Elhadad, 2010;

ABAE
Intro & motivation

Aspect extraction is one of the key tasks in sentiment analysis. It aims to extract entity aspects on which opinions have been expressed (Hu and Liu, 2004; Liu, 2012). For example, in the sentence *"The beef was tender and melted in my mouth"*, the aspect term is *"beef"*. Two sub-tasks are performed in aspect extraction: (1) extracting all aspect terms (e.g., *"beef"*) from a review corpus, (2) clustering aspect terms with similar meaning into categories where each category represents a single

ABAE
Intro & motivation

butions over word types. While the mixture of aspects discovered by LDA-based models may describe a corpus fairly well, we find that the individual aspects inferred are of poor quality – aspects often consist of unrelated or loosely-related concepts. This may substantially reduce users' con-

ity. Conventional LDA models do not directly encode word co-occurrence statistics which are the primary source of information to preserve topic coherence (Mimno et al., 2011). They implicitly capture such patterns by modeling word generation from the document level, assuming that each word is generated independently. Furthermore, LDA-based models need to estimate a distribution of topics for each document. Review documents tend to be short, thus making the estimation of topic distributions more difficult.

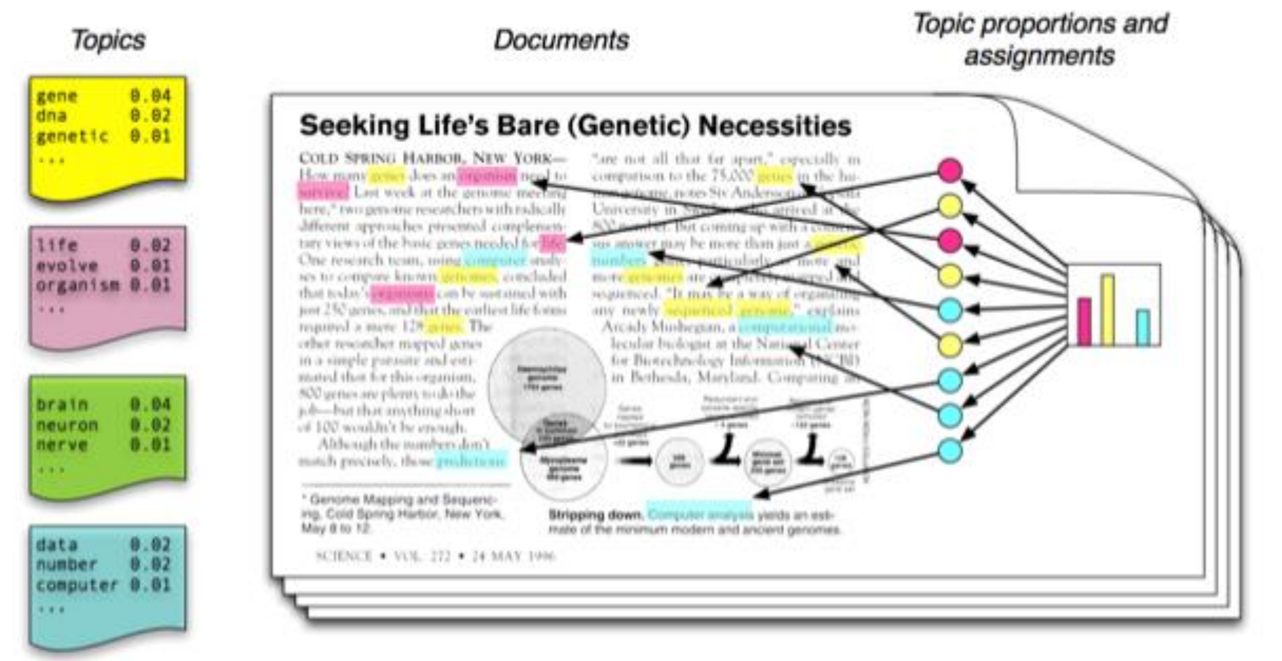


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

ABAE
Intro & motivation

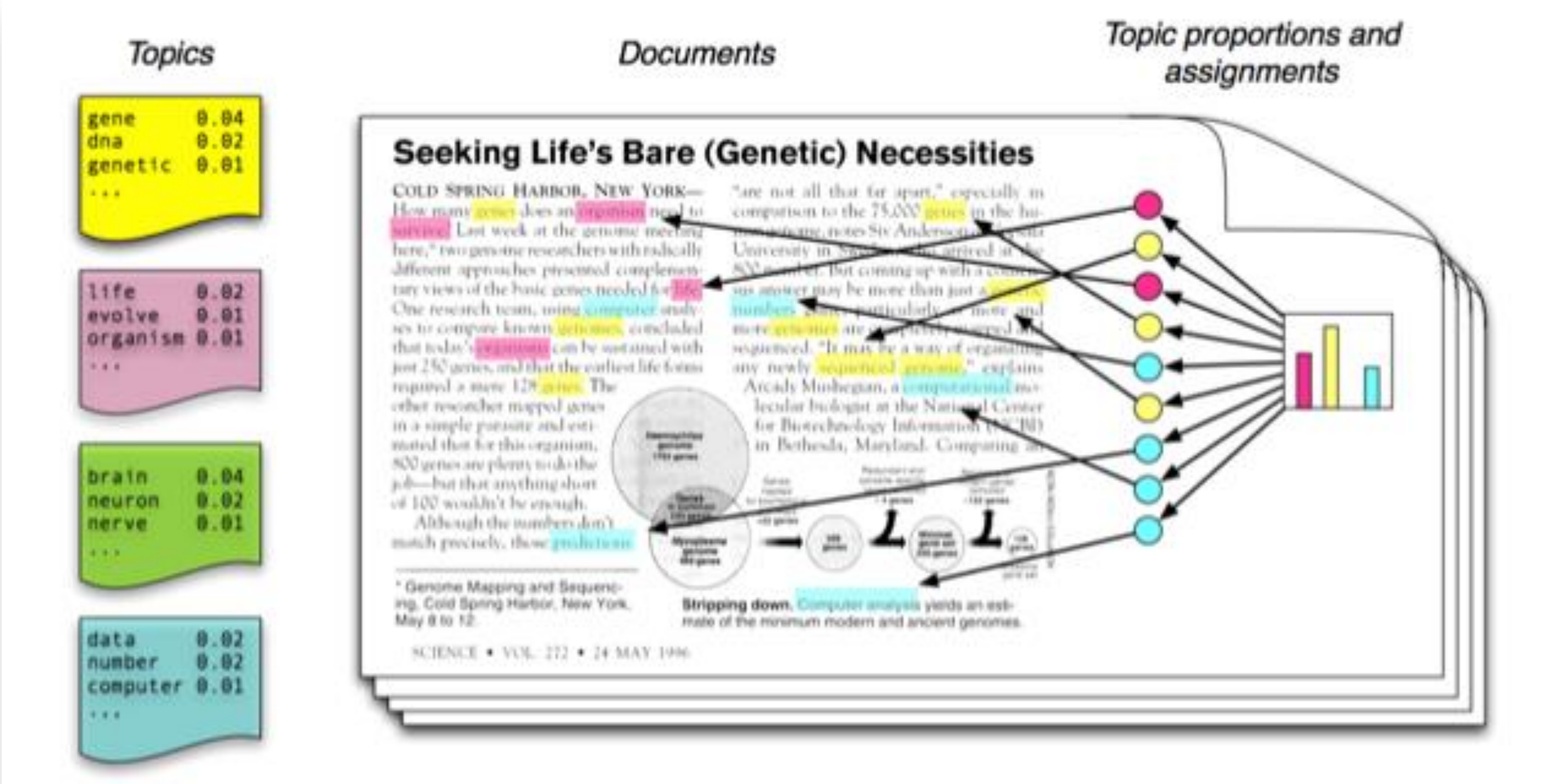


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

ABAE Approach

- Topic words must be the most important words one needs to capture the meaning a sentence => they deserve highest attention (think content words from a linguistics viewpoint)

- Word2vec models already capture word co-occurrence, we can use that for grouping topic words into categories.

- It follows then that if I:

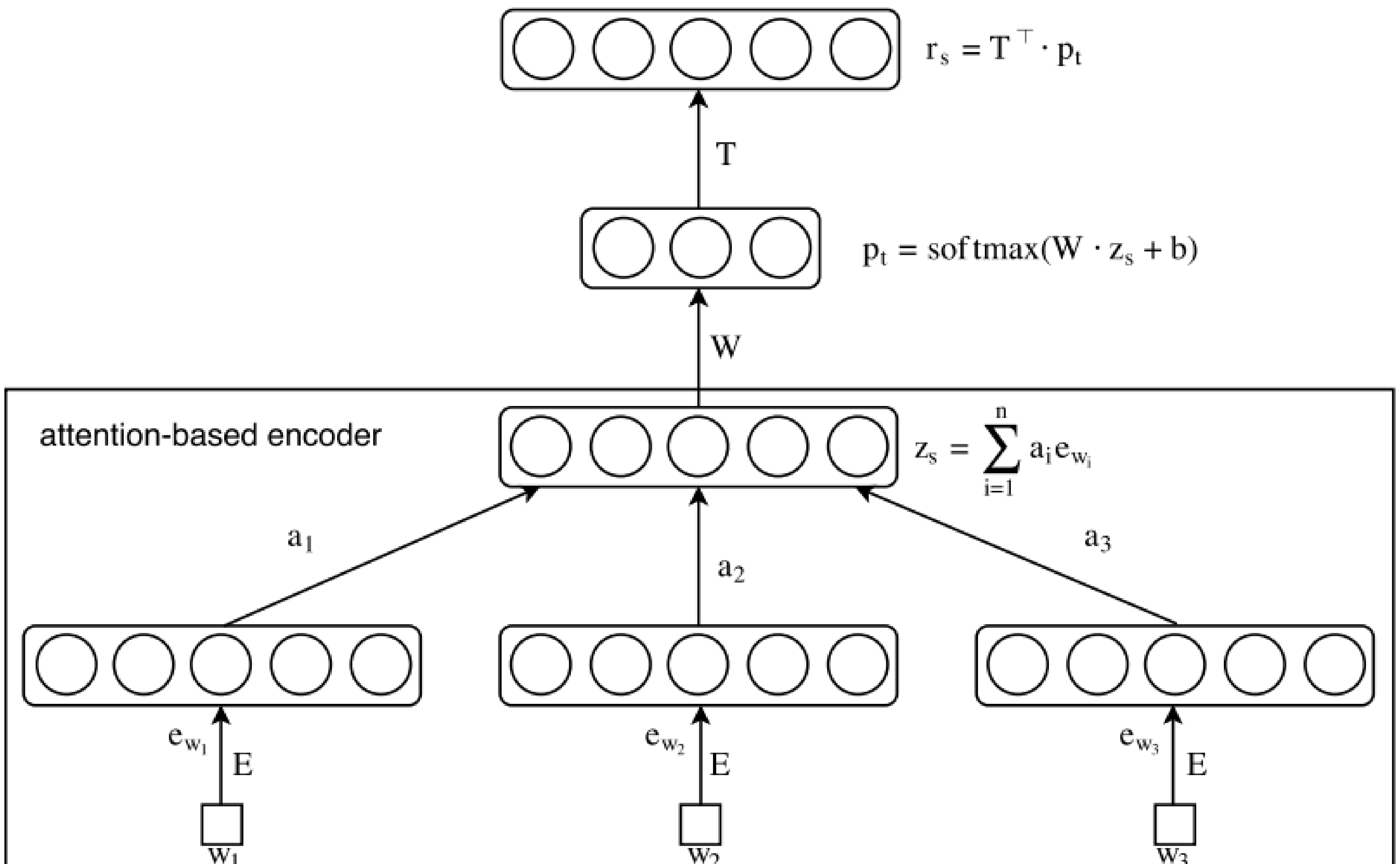
  - choose the number of topics I am interested in;

  - Use offline trained embeddings;

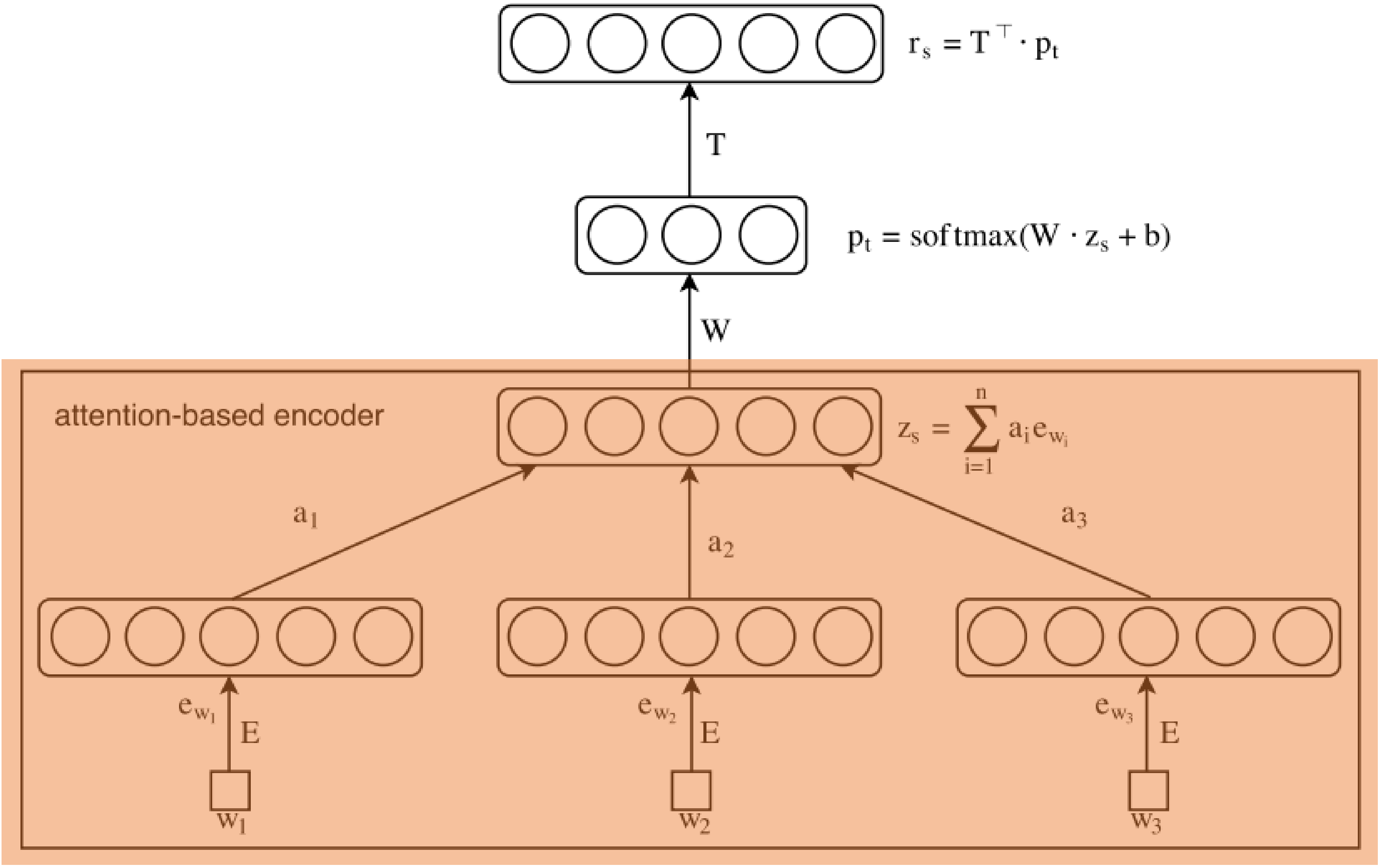
  - Use an autoencoder to reconstruct a sentence embedding

  The middle layer of the autoencoder should give me the list of topic embeddings and I can infer the actual words by kNN and the topic categories by doing clustering.

ABAE Architecture

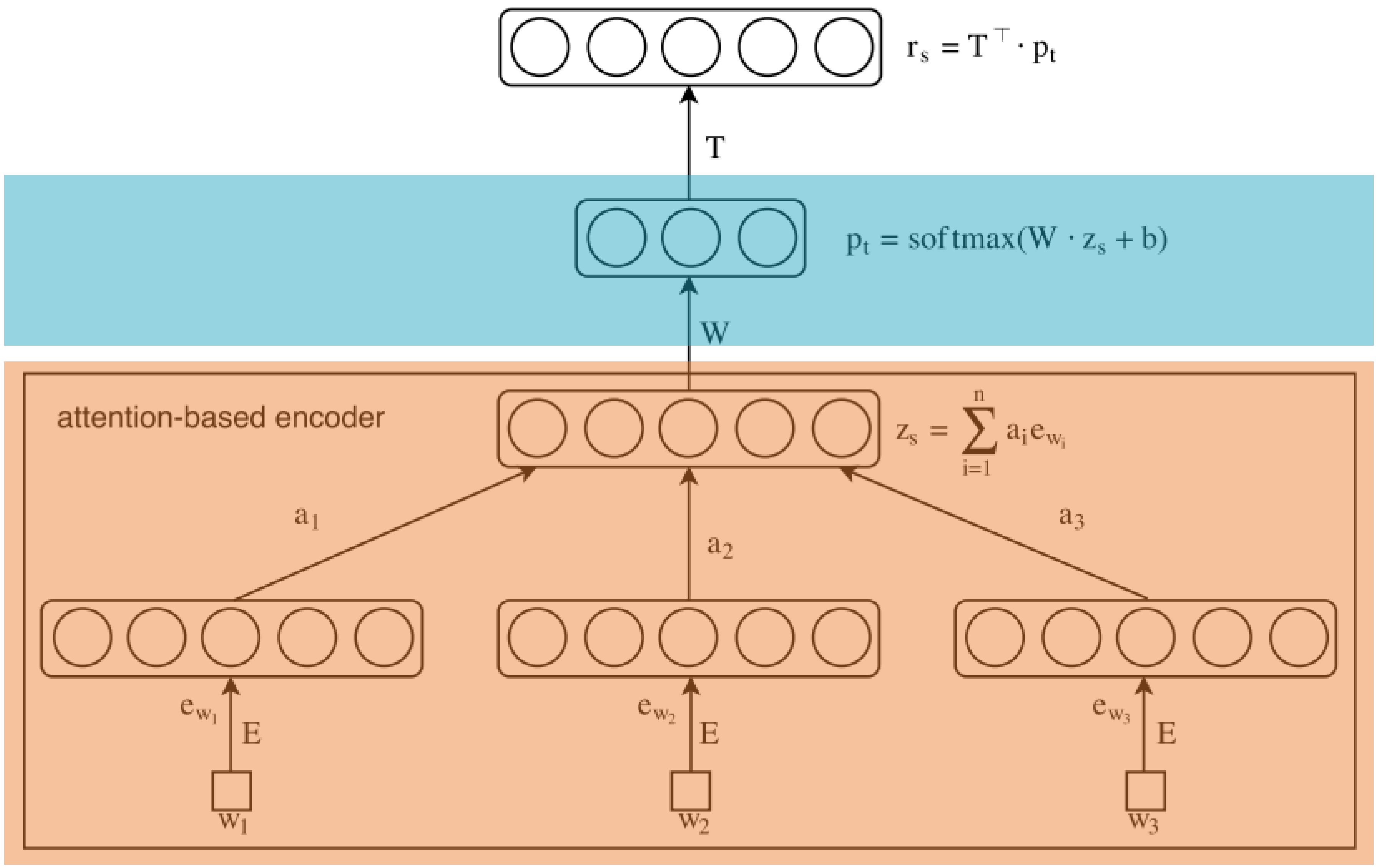


ABAE Architecture



Build sentence embedding/representation from words

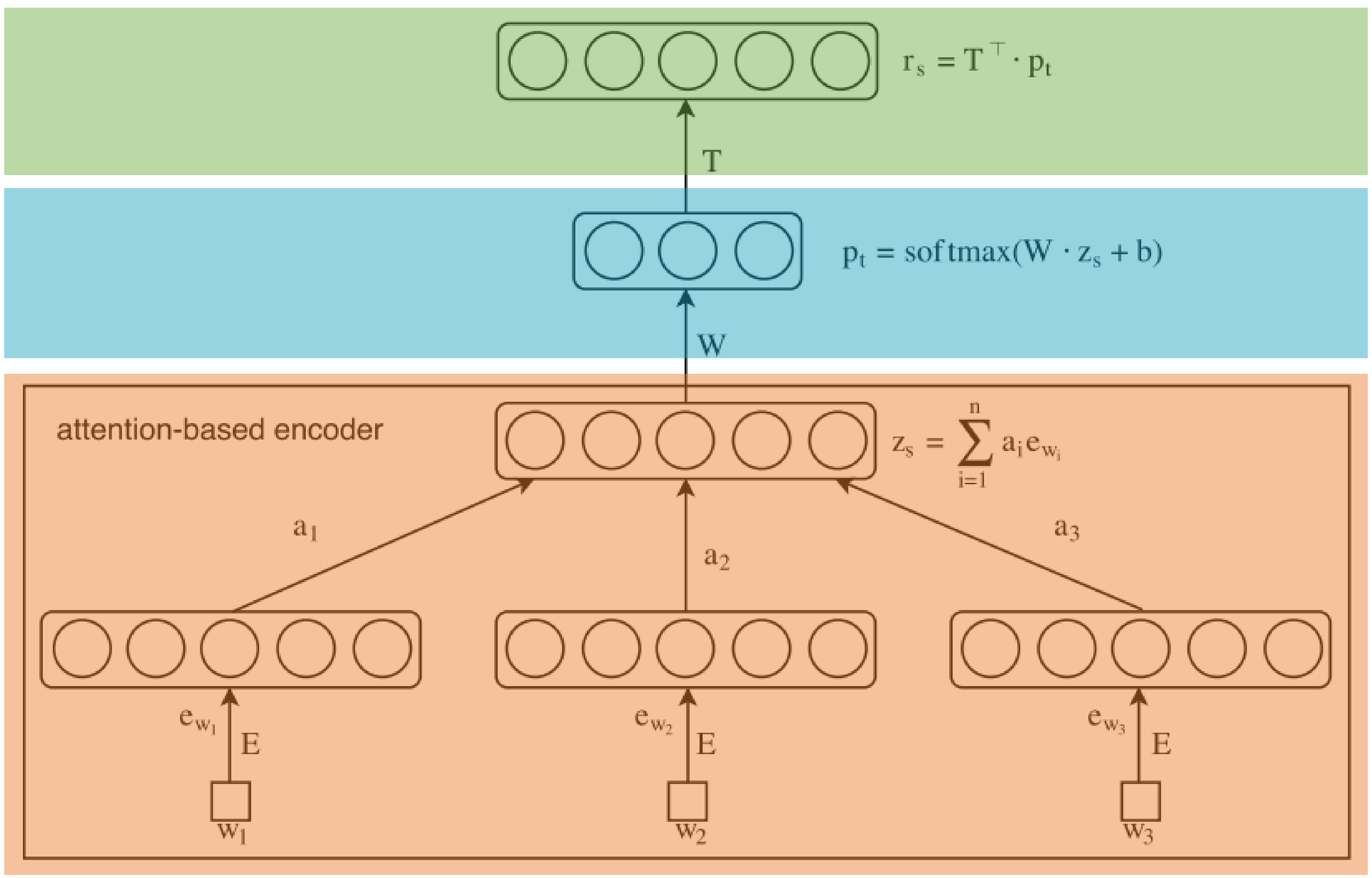
ABAE Architecture



Represent sentence as scores assigned to topic words in T

Build sentence embedding/representation from words

ABAE Architecture

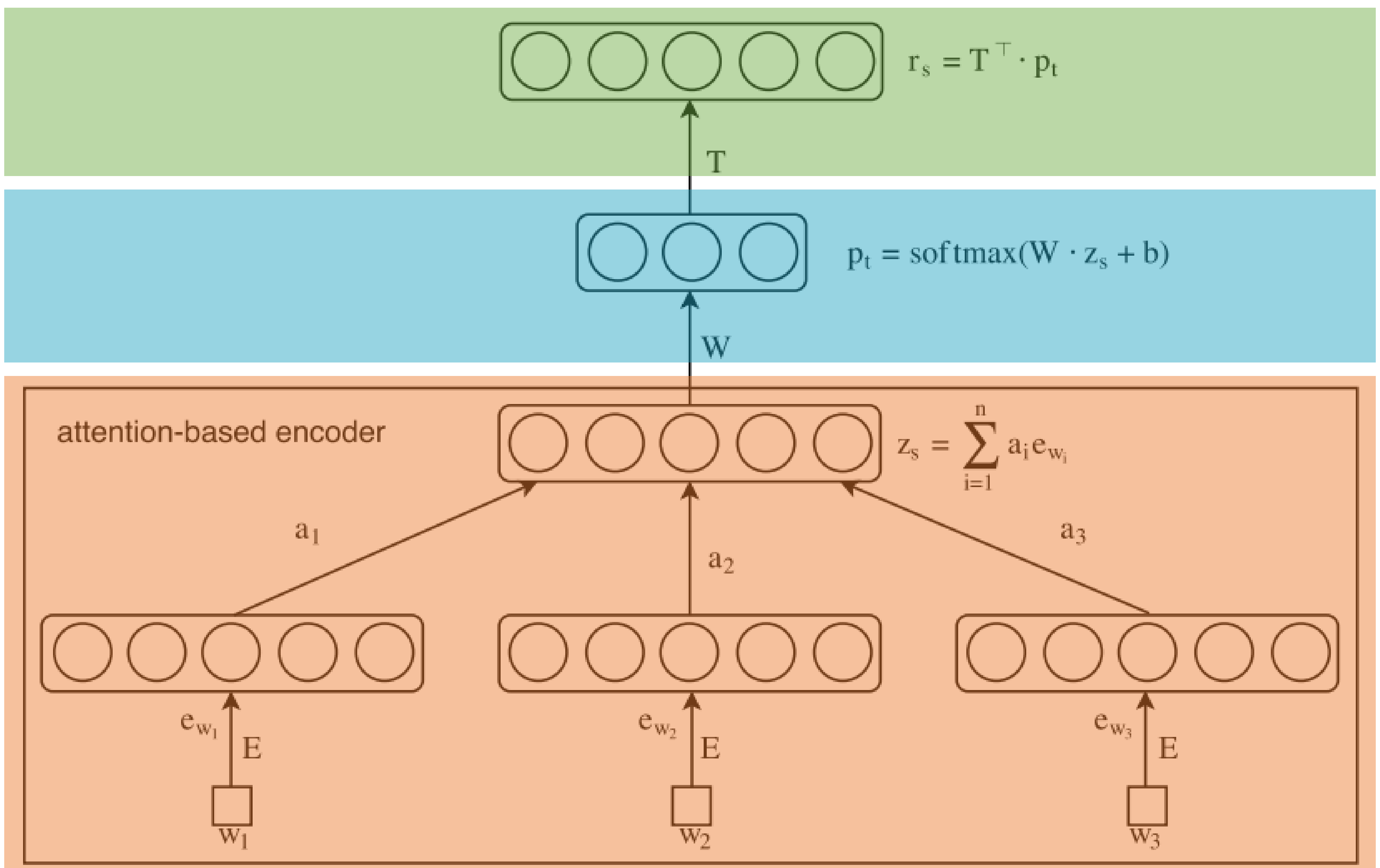


<u>**Reconstruct**</u>
Get original sentence representation from topic

Represent sentence as scores assigned to topic words in T

Build sentence embedding/representation from words

ABAE Architecture



$$a_i = \frac{\exp(d_i)}{\sum_{j=1}^{n} \exp(d_j)}$$

$$d_i = \mathbf{e}_{w_i}^\top \cdot \mathbf{M} \cdot \mathbf{y}_s$$

$$\mathbf{y}_s = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_{w_i}$$

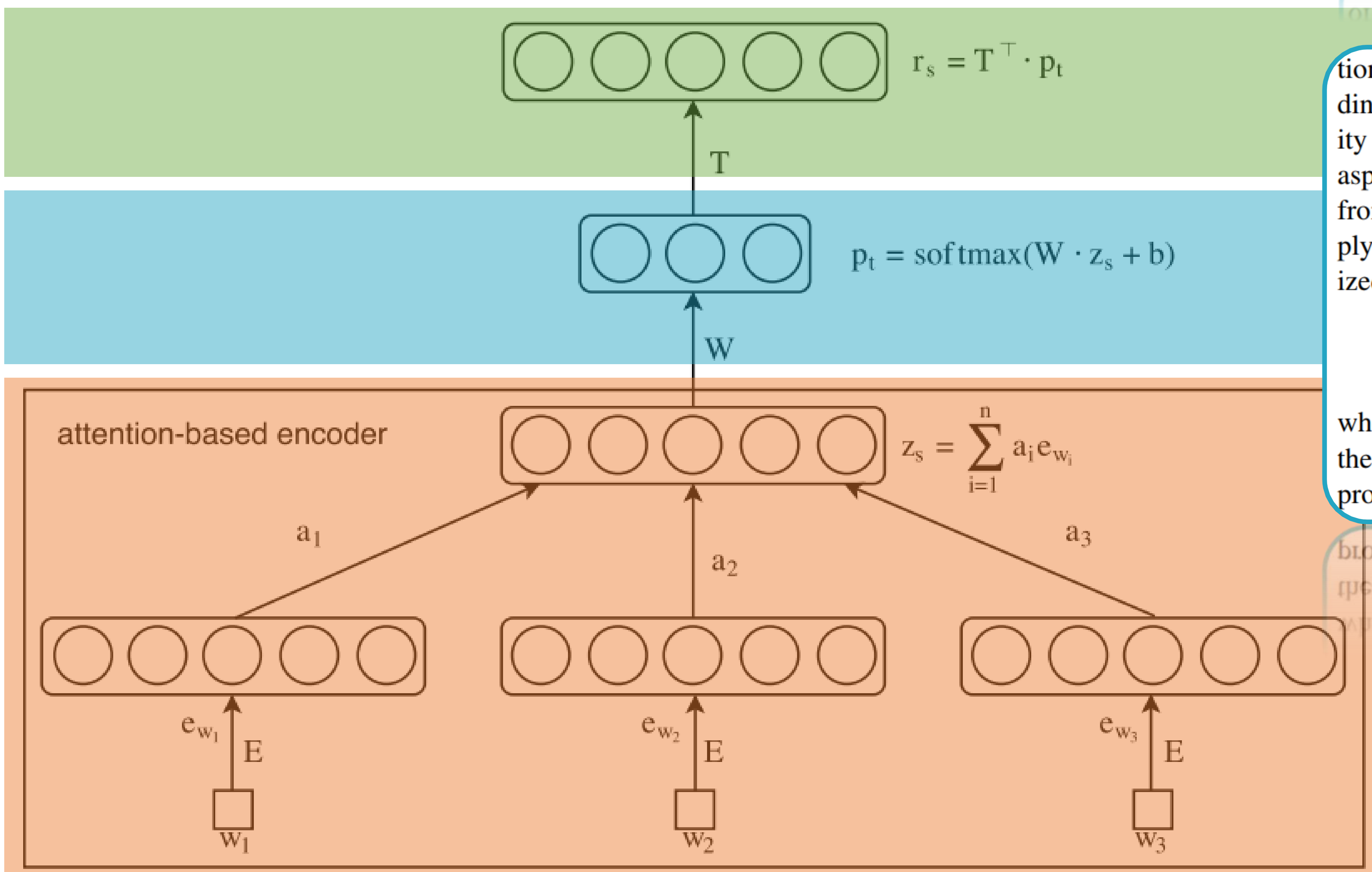
ABAE Architecture

$$a_i = \frac{\exp(d_i)}{\sum_{j=1}^{n} \exp(d_j)}$$

$$d_i = \mathbf{e}_{w_i}^{\top} \cdot \mathbf{M} \cdot \mathbf{y}_s$$

$$\mathbf{y}_s = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_{w_i}$$

ABAE Architecture



space with words. This requires an aspect embedding matrix $\mathbf{T} \in \mathbb{R}^{K \times d}$, where K, the number of aspects defined, is much smaller than V. The

tion, $\mathbf{p}_t$ is the weight vector over K aspect embeddings, where each weight represents the probability that the input sentence belongs to the related aspect. $\mathbf{p}_t$ can simply be obtained by reducing $\mathbf{z}_s$ from d dimensions to K dimensions and then applying a softmax non-linearity that yields normalized non-negative weights:

$$\mathbf{p}_t = softmax(\mathbf{W} \cdot \mathbf{z}_s + \mathbf{b}) \tag{6}$$

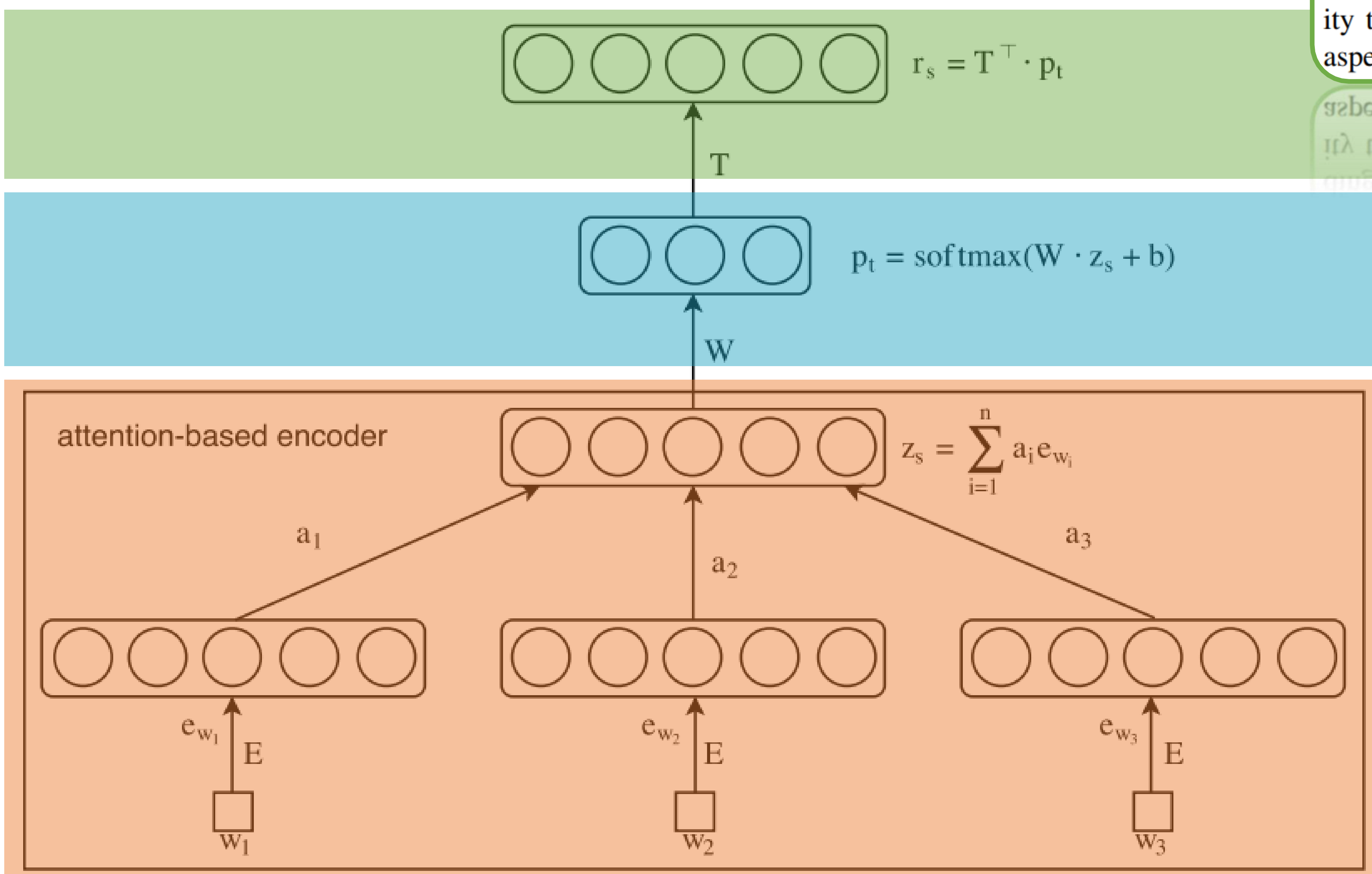
where $\mathbf{W}$, the weighted matrix parameter, and $\mathbf{b}$, the bias vector, are learned as part of the training process.

ABAE Architecture

transitions, which is similar to an autoencoder. Intuitively, we can think of the reconstruction as a linear combination of aspect embeddings from $\mathbf{T}$:

$$\mathbf{r}_s = \mathbf{T}^\top \cdot \mathbf{p}_t \tag{5}$$

where $\mathbf{r}_s$ is the reconstructed vector representation, $\mathbf{p}_t$ is the weight vector over K aspect embeddings, where each weight represents the probability that the input sentence belongs to the related aspect. $\mathbf{p}_t$ can simply be obtained by reducing $\mathbf{z}_s$



ABAE
Loss function

$$L(\theta) = J(\theta) + \lambda U(\theta)$$

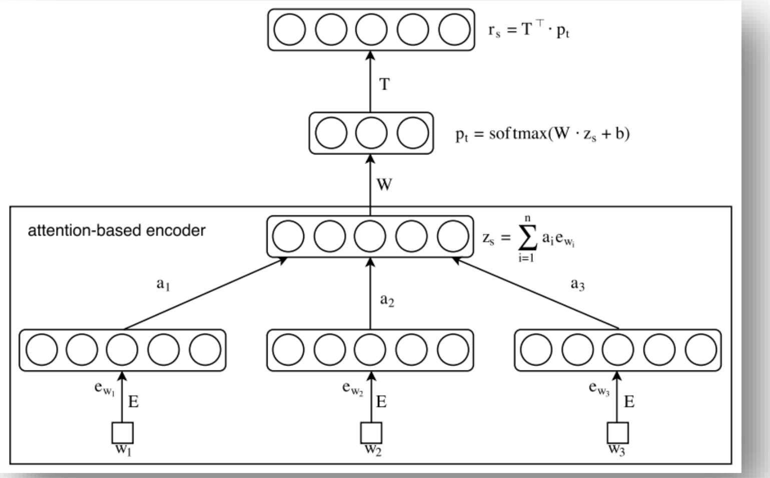
Contrastive max-margin

$$J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} \max(0, 1 - \mathbf{r}_s \mathbf{z}_s + \mathbf{r}_s \mathbf{n}_i)$$

Marg in

Positive sample

Negative samples



ABAE
Loss function

$$L(\theta) = J(\theta) + \lambda U(\theta)$$

Contrastive max-margin

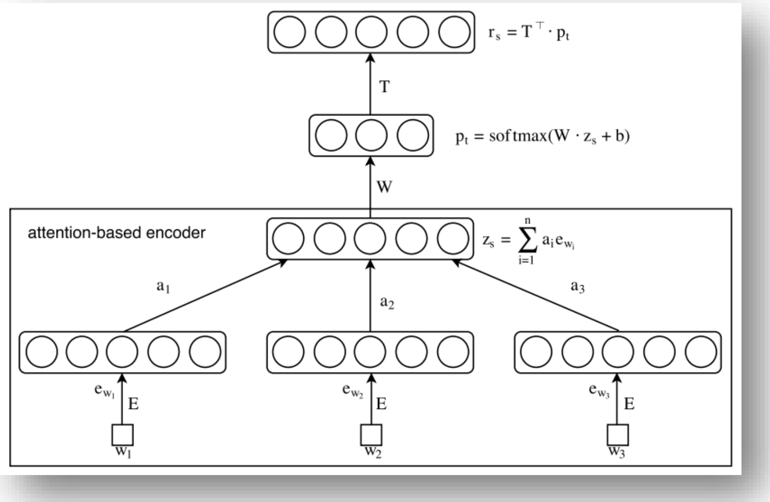
$$J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} \max(0, 1 - \mathbf{r}_s \mathbf{z}_s + \mathbf{r}_s \mathbf{n}_i)$$

$$U(\theta) = \| \mathbf{T}_n \cdot \mathbf{T}_n^\top - \mathbf{I} \|$$

Marg in

Positive sample

Negative samples



ABAE
Loss function

$$L(\theta) = J(\theta) + \lambda U(\theta)$$

fer from redundancy problems during training. To ensure the diversity of the resulting aspect embeddings, we add a regularization term to the objective function J to encourage the uniqueness of each aspect embedding:

where $\mathbf{I}$ is the identity matrix, and $\mathbf{T}_n$ is $\mathbf{T}$ with each row normalized to have length 1. Any non-

pect embeddings. U reaches its minimum value when the dot product between any two different aspect embeddings is zero. Thus the regularization

Contrastive max-margin

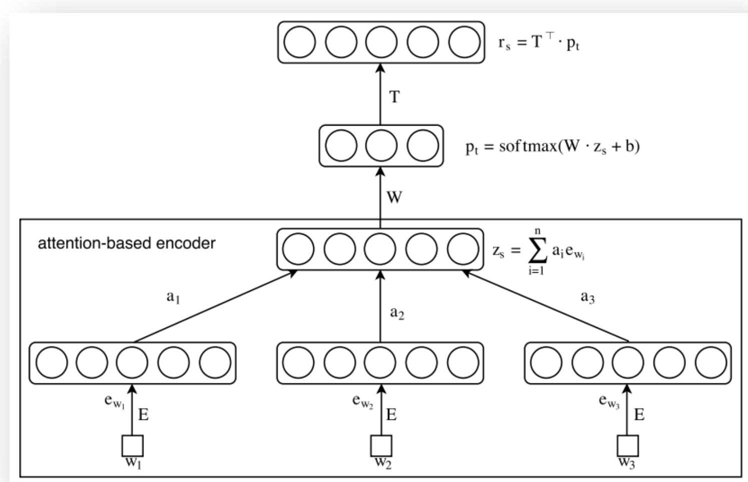
$$\boxed{J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} \max(0, 1 - \mathbf{r}_s \mathbf{z}_s + \mathbf{r}_s \mathbf{n}_i)}$$

$$\boxed{U(\theta) = \|\mathbf{T}_n \cdot \mathbf{T}_n^\top - \mathbf{I}\|}$$

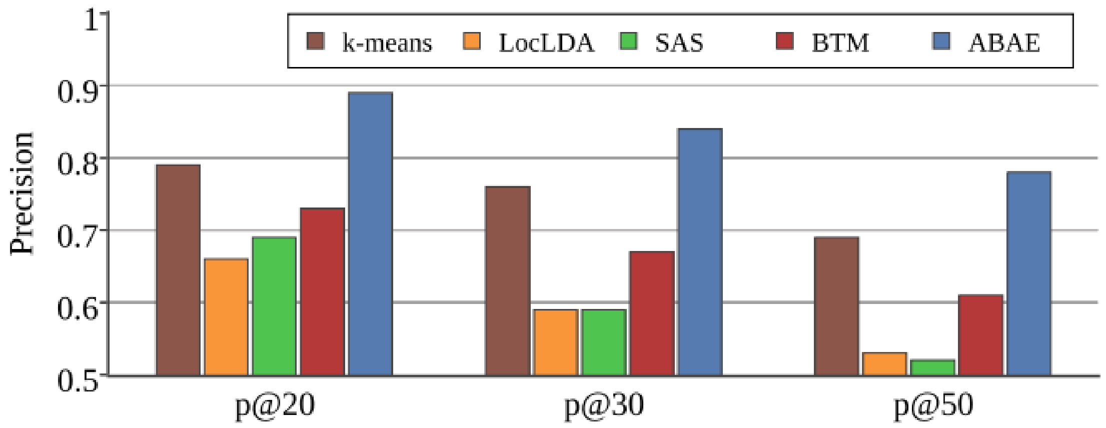
Marg in

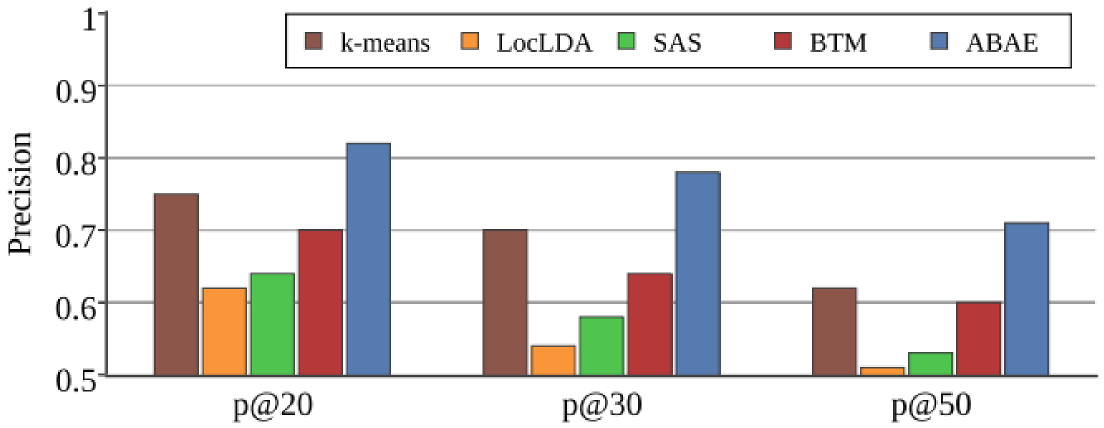
Positive sample

Negative samples

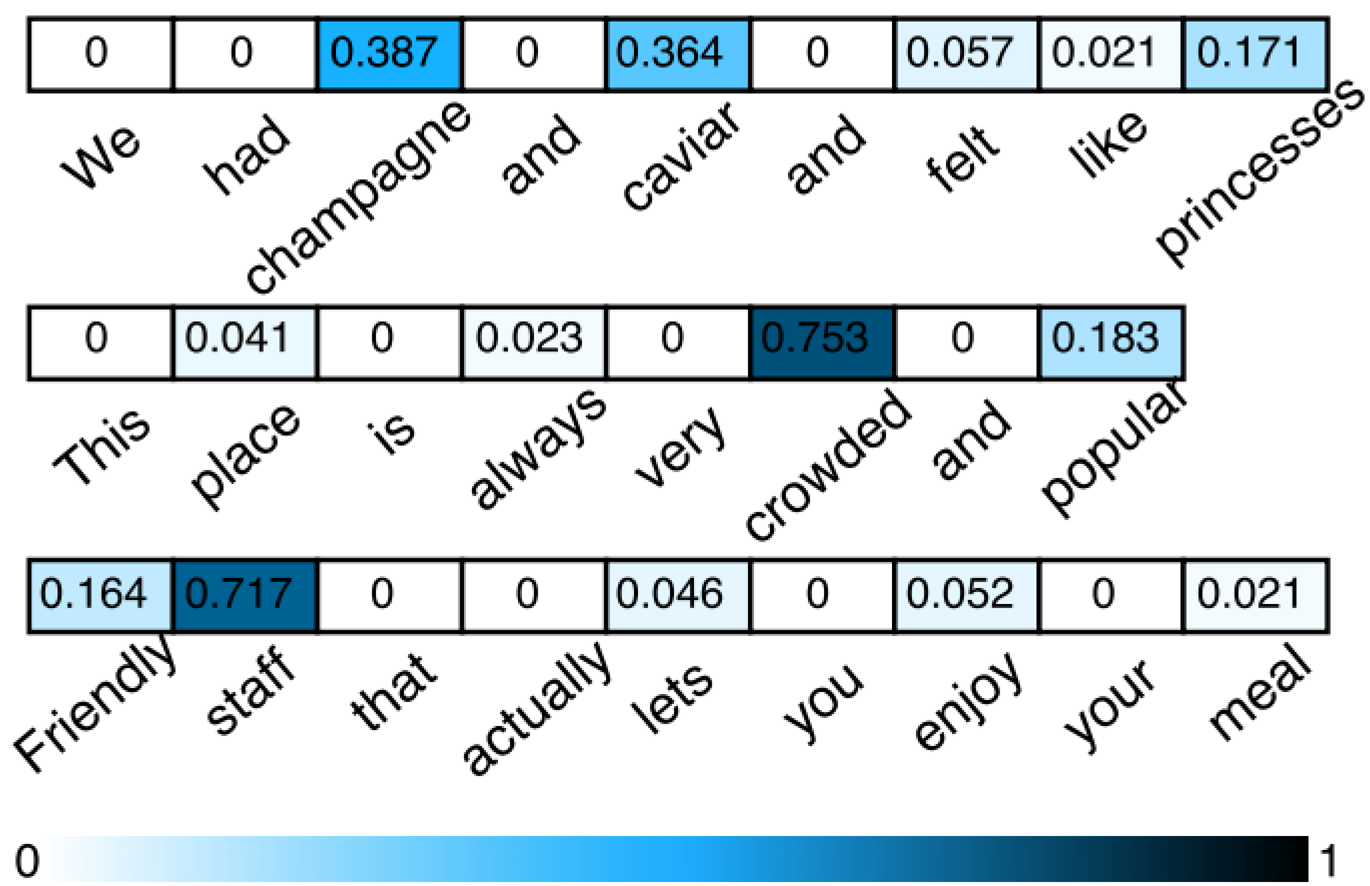


ABAE Results





ABAE Results



ABAE
Back to self attention

How else can we do this?

$$a_i = \frac{\exp(d_i)}{\sum_{j=1}^{n} \exp(d_j)}$$

$$d_i = \mathbf{e}_{w_i}^{\top} \cdot \mathbf{M} \cdot \mathbf{y}_s$$

$$\mathbf{y}_s = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_{w_i}$$