Reminders

HOMEWORK 9 OR PROJECT MILESTONE 1 ARE DUE TONIGHT BY 11:59PM

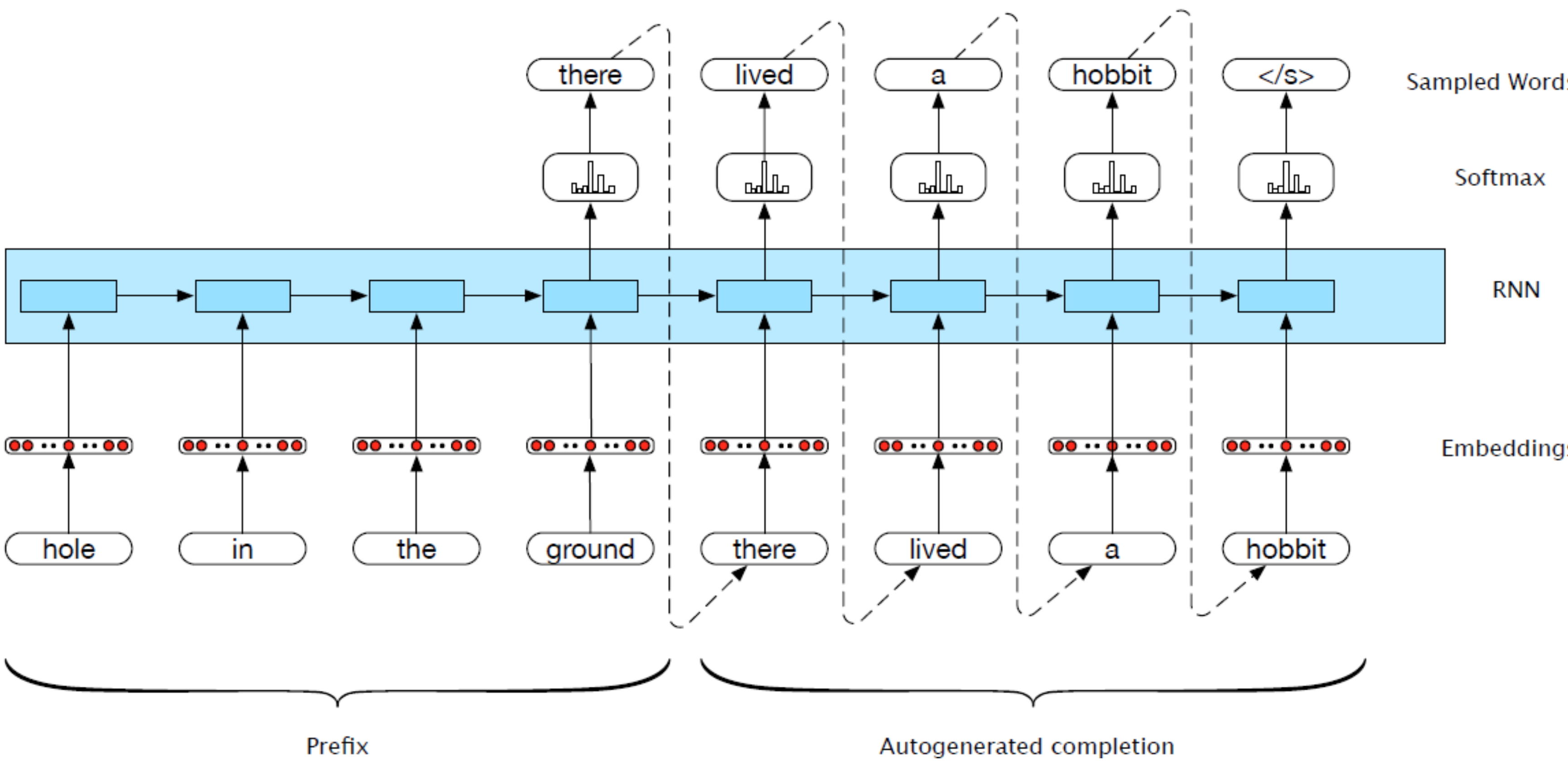
QUIZ ON CHAPTER 28 IS DUE MONDAY.

HW10 ON NEURAL MACHINE TRANSLATION WILL BE RELEASED SOON. MILESTONE 2 IS READY.

Encoder-Decoder Models

JURAFSKY AND MARTIN CHAPTER 10

Generation with prefix

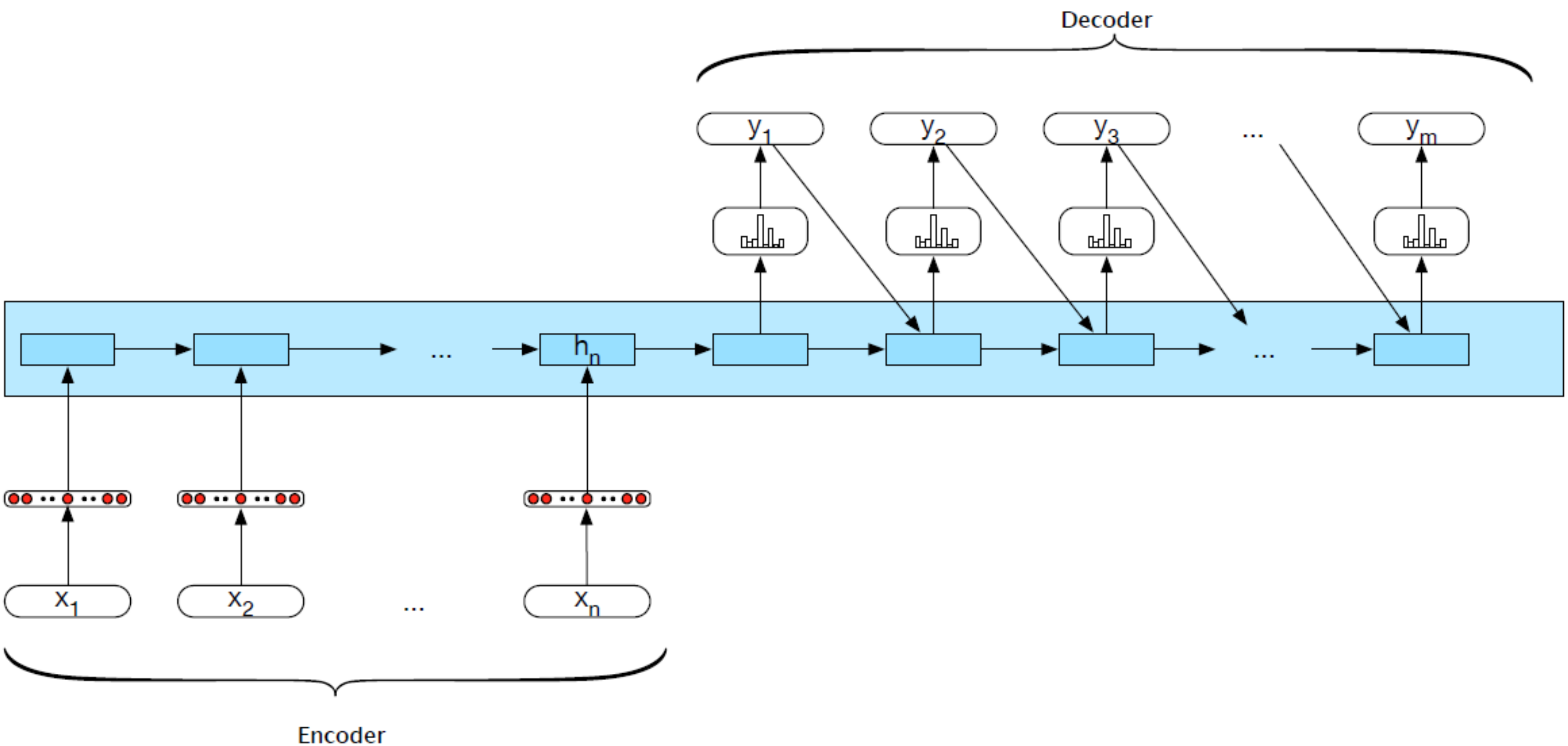


Encoder-Decoder Networks

We can abstract away from the task of MT to talk about the general **encoder-decoder architecture:**

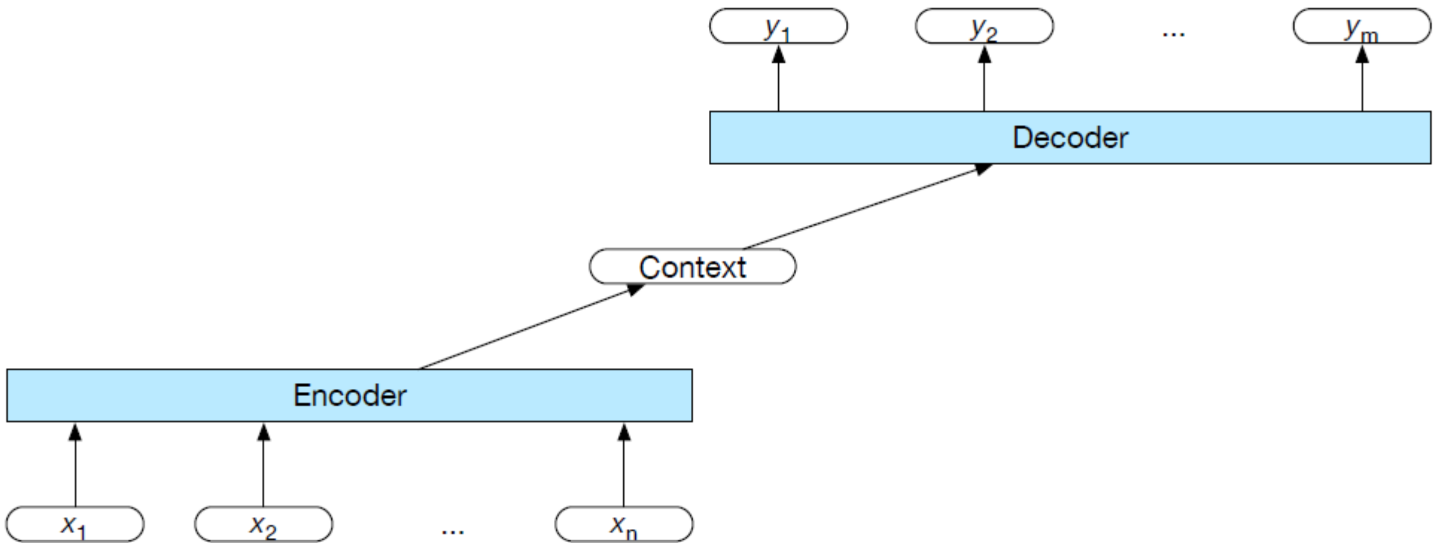
1. An **encoder** takes an input sequence x^n_1, and generates a corresponding sequence of contextualized representations, h^n_1.
2. A **context vector**, *c*, is a function of h^n_1, and conveys the essence of the input to the decoder.
3. A **decoder** accepts *c* as input and generates an arbitrary length sequence of hidden states h^m_1 , from which can be used to create a corresponding sequence of output states y^m_1 .

Encoder-decoder networks



Encoder-decoder networks

- An encoder that accepts an input sequence and generates a corresponding sequence of contextualized representations
- A context vector that conveys the essence of the input to the decoder
- A decoder, which accepts context vector as input and generates an arbitrary length sequence of hidden states, from which a corresponding sequence of output states can be obtained

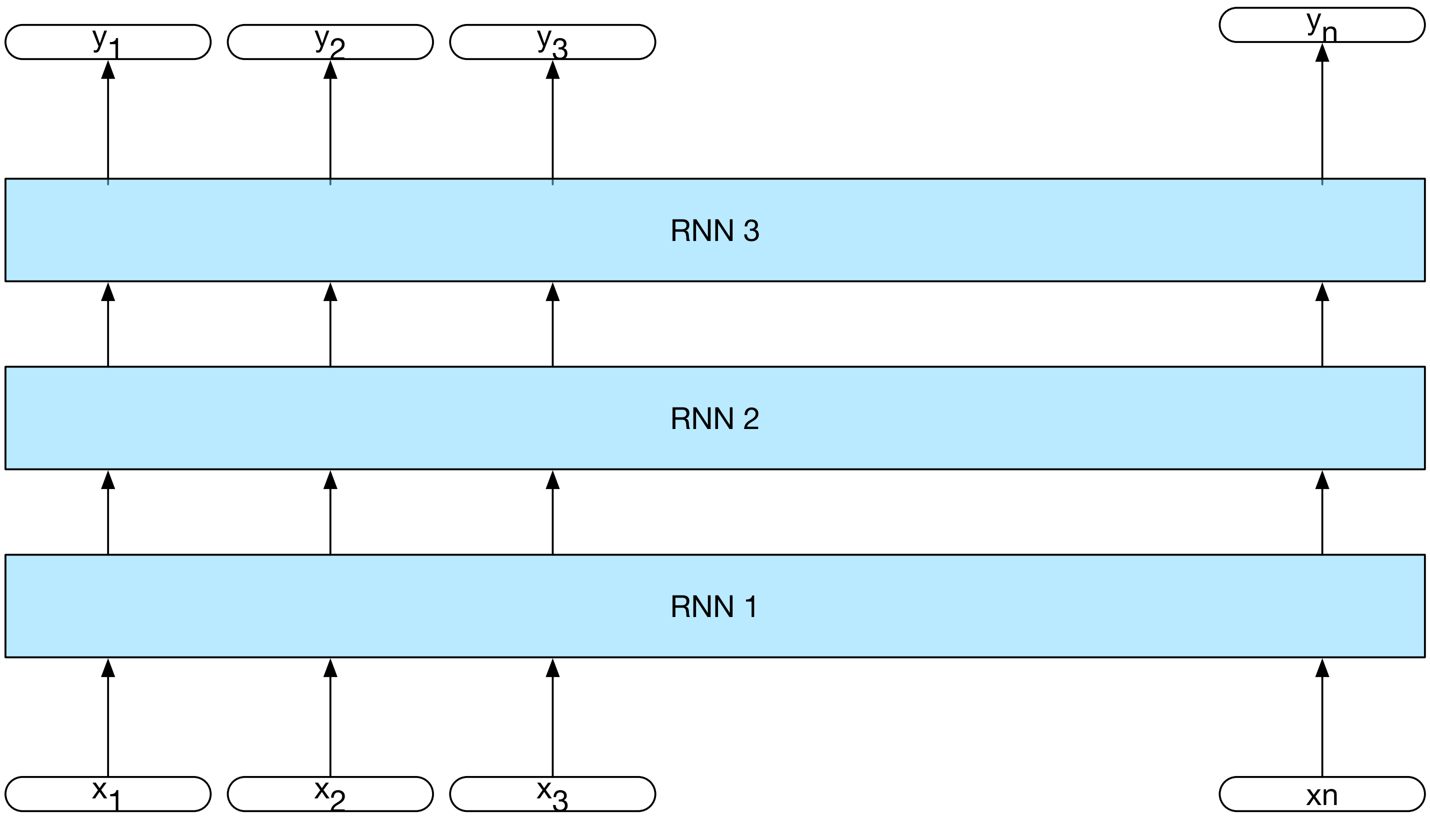


Encoder

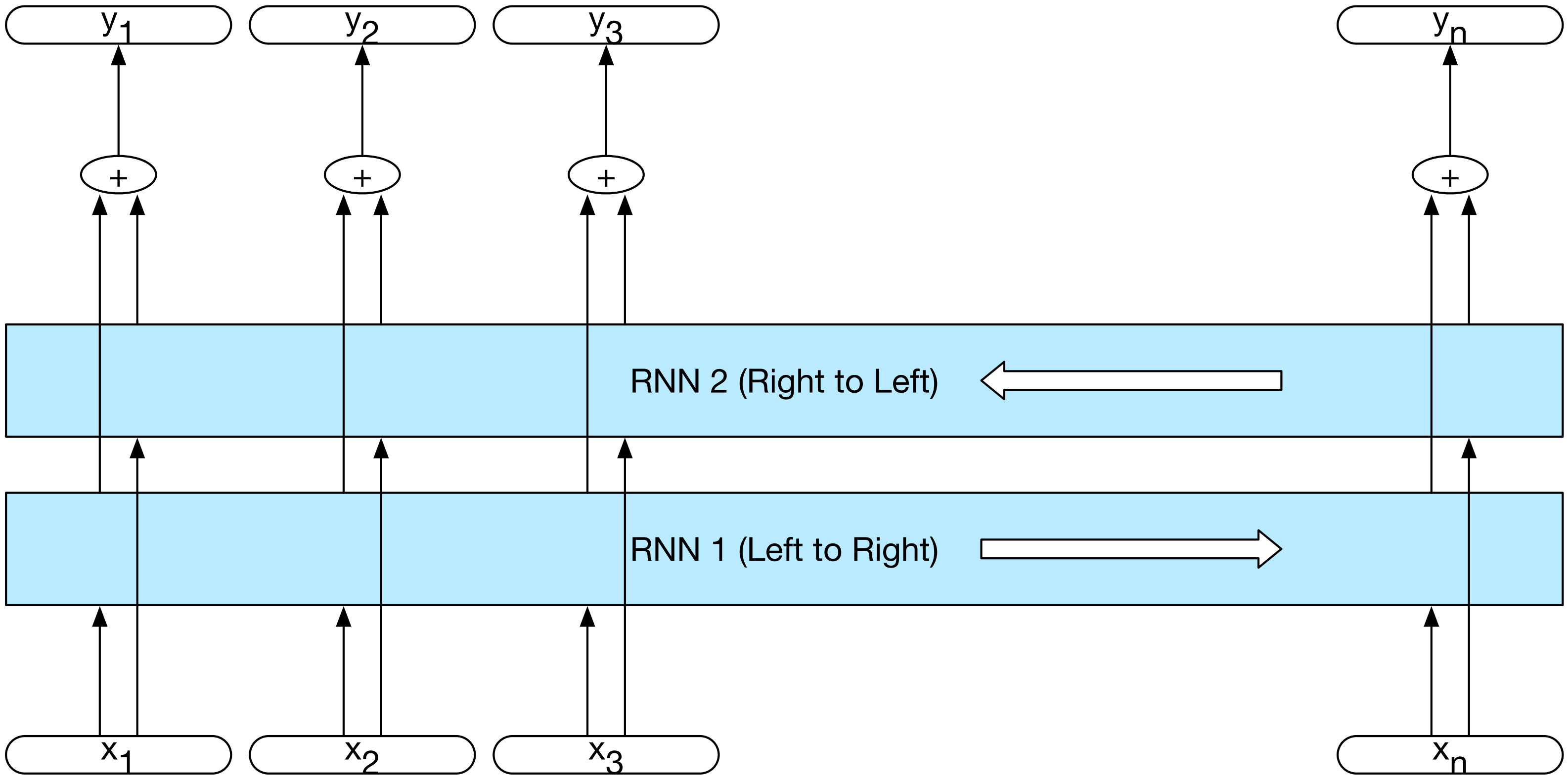
Pretty much any kind of RNN or its variants can be used as an encoder. Researchers have used simple RNNs, LSTMs, GRUs, or even convolutional networks.

A widely used encoder design makes use of stacked Bi-LSTMs where the hidden states from top layers from the forward and backward passes are concatenated

Stacked RNNs



Bidirectional RNNs



Decoder

For the decoder, autoregressive generation is used to produce an output sequence, an element at a time, until an end-of-sequence marker is generated.

This incremental process is guided by the context provided by the encoder as well as any items generated for earlier states by the decoder.

Encoder

$$\begin{aligned} c &= h_n^e \\ h_0^d &= c \end{aligned}$$

Decoder

$$\begin{aligned} h_t^d &= g(\hat{y}_{t-1}, h_{t-1}^d) \\ z_t &= f(h_t^d) \\ y_t &= \text{softmax}(z_t) \end{aligned}$$

Decoder Weaknesses

In early encoder-decoder approaches, the context vector *c* was only directly available at the beginning of the generation process.

This meant that its influence became less-and-less imporant as the output sequence was generated.

One solution is to make *c* available at each step in the decoding process, when generating the hidden states in the deocoder

$$h_t^d = g(\hat{y}_{t-1}, h_{t-1}^d, c)$$

and while producing the generated output.

$$y_t = \text{softmax}(\hat{y}_{t-1}, z_t, c)$$

Choosing the best output

For neural generation, where we are trying to generate novel outputs, we can simply sample from the softmax distribution.

In MT where we're looking for a specific output sequence, sampling isn't appropriate and would likely lead to some strange output.

Instead we choose the most likely output at each time step by taking the argmax over the softmax output

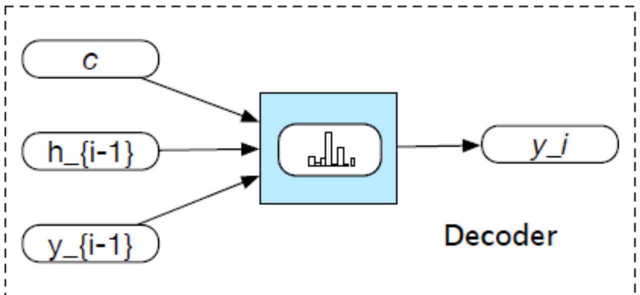
$$\hat{y} = \text{argmax} P(y_i | y_< i)$$

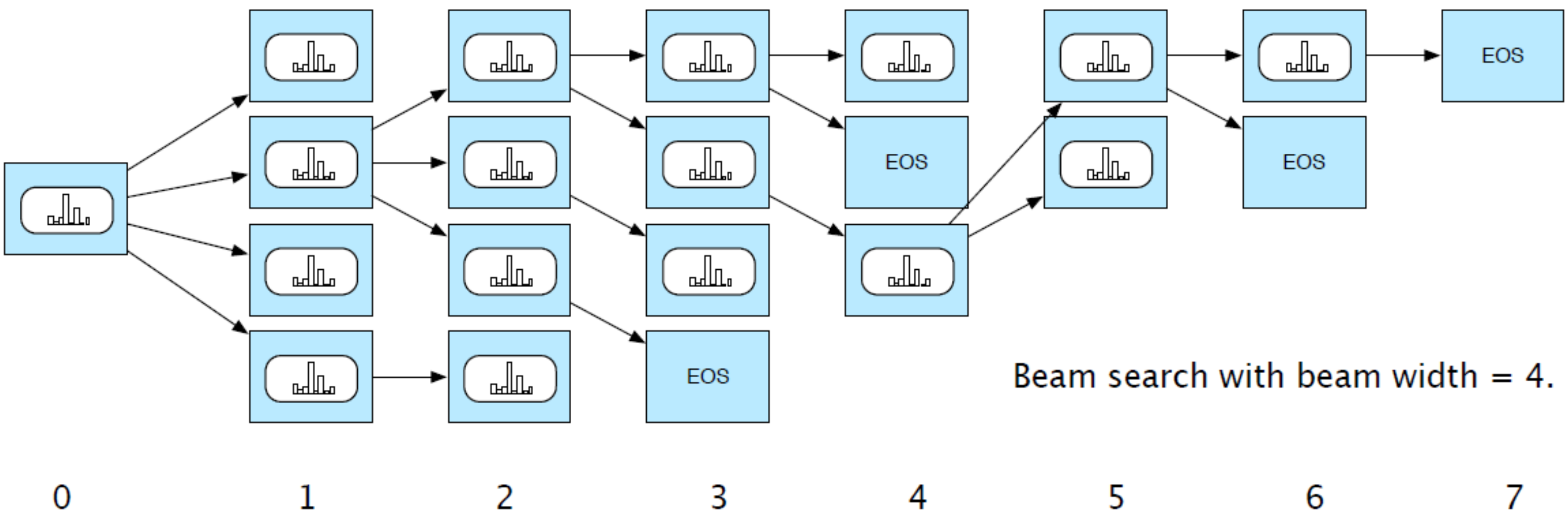
Beam search

In order to systematically explore the space of possible outputs for applications like MT, we need to control the exponential growth of the search space.

Beam search: combining a breadth-first-search strategy with a heuristic filter that scores each option and prunes the search space to stay within a fixed-size memory footprint, called the beam width

Beam search





Beam search with beam width = 4.

Attention

Weaknesses of the context vector:

- Only directly available at the beginning of the process and its influence will wane as the output sequence is generated
- Context vector is a function (e.g. last, average, max, concatenation) of the hidden states of the encoder. This approach loses useful information about each of the individual encoder states

Potential solution: **attention mechanism**

Attention

- Replace the static context vector with one that is dynamically derived from the encoder hidden states at each point during decoding
- A new context vector is generated at each decoding step and takes all encoder hidden states into derivation
- This context vector is available to decoder hidden state calculations

$$h_i^d = g\left(\hat{y}_{i-1}, h_{i-1}^d, c_i\right)$$

Attention

- To calculate c_i, first find relevance of each encoder hidden state to the decoder state. Call it $score(h_{i-1}^{d}, h_j^{e})$ for each encoder state j
- The $score$ can simply be dot product,

$$score(h_{i-1}^{d}, h_j^{e}) \;=\; h_{i-1}^{d} \cdot h_j^{e}$$

Attention

- The score can also be parameterized with weights

$$score(h_{i-1}^{d}, h_{j}^{e}) \;=\; h_{t-1}^{d} W_s h_{j}^{e}$$

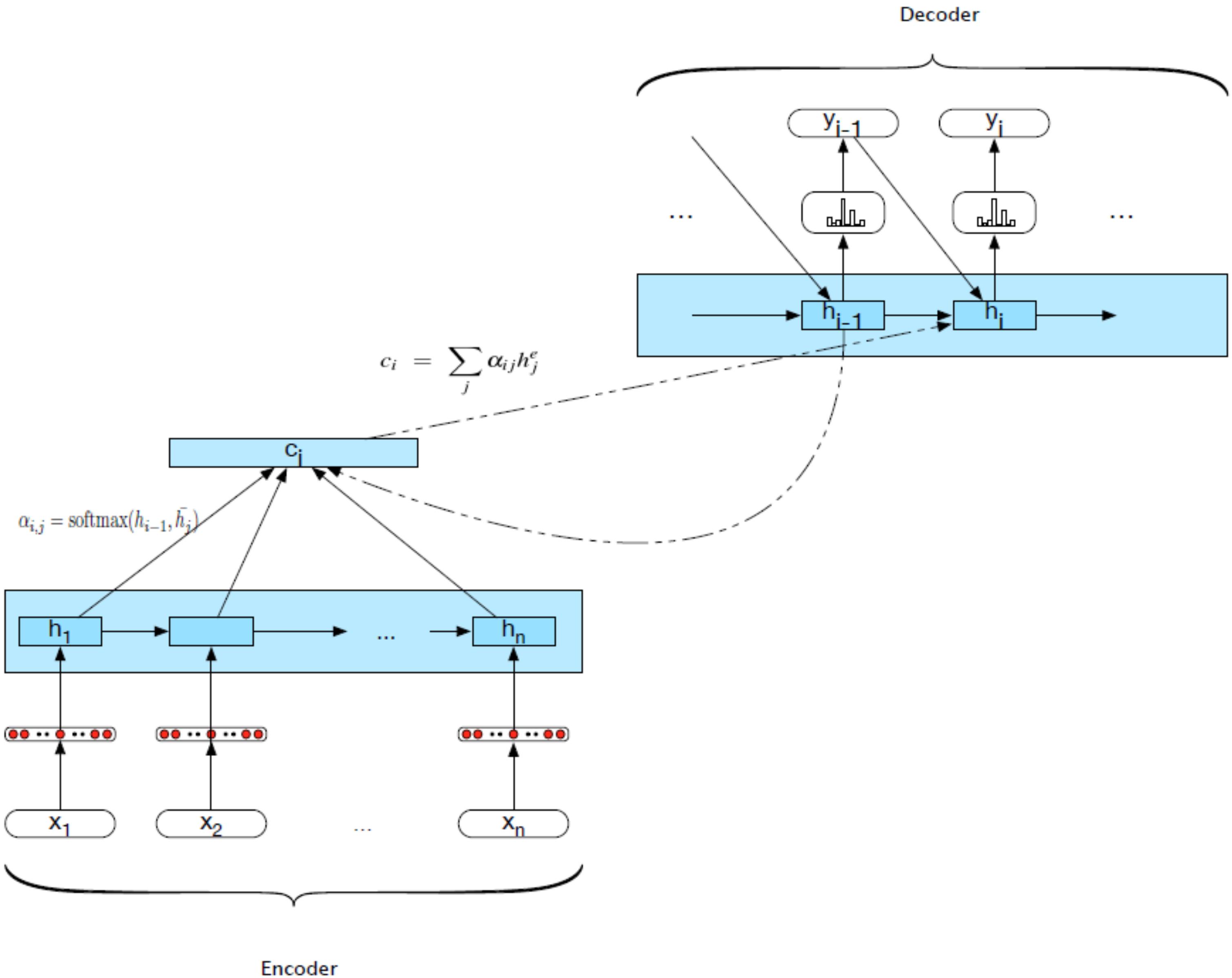
- Normalize them with a softmax to create a vector of weights $\alpha_{i,j}$ that tells us the proportional relevance of each encoder hidden state j to the current decoder state i

$$\alpha_{i,j} = softmax(score\left(h_{i-1}^{d}, h_{j}^{e}\right) \forall j \in e)$$

- Finally, context vector is the weighted average of encoder hidden states

$$c_i = \sum_{j} \alpha_{i,j}\, h_{j}^{e}$$

Attention mechanism



Applications of Encoder-Decoder Networks

- Text summarization
- Text simplification
- Question answering
- Image captioning
- And more. What do those tasks have in common?

Neural Machine Translation

SLIDES FROM GRAHAM NEUBIG, CMU

[NEURAL MACHINE TRANSLATION AND SEQUENCE-TO-SEQUENCE MODELS: A TUTORIAL]

Machine Translation

Translation from one language to another

I'm giving a talk at University of Pennsylvania

↓

ペンシルベニア大学で講演をしています。

Long-distance Dependencies

Agreement in number, gender, etc.

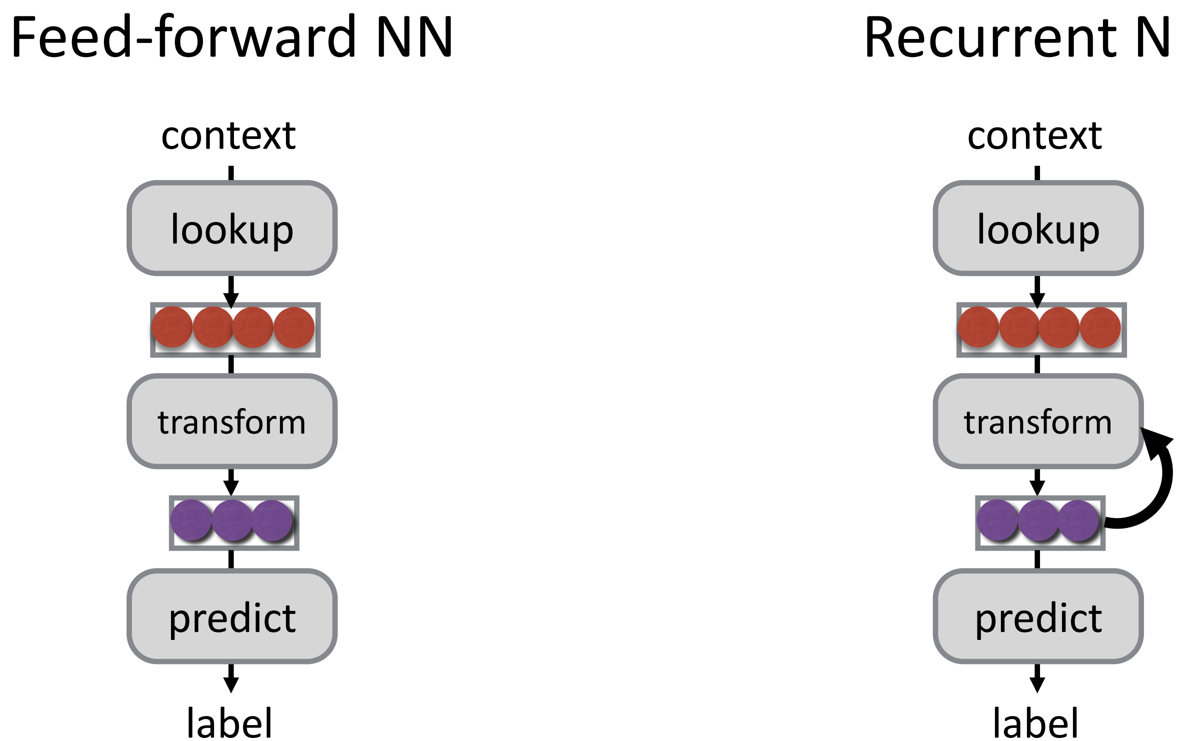
1. **He** does not have very much confidence in **himself**.
2. **She** does not have very much confidence in **herself**.

Selectional preference:

1. The **reign** has lasted as long as the life of the **queen**.
2. The **rain** has lasted as long as the life of the **clouds**.

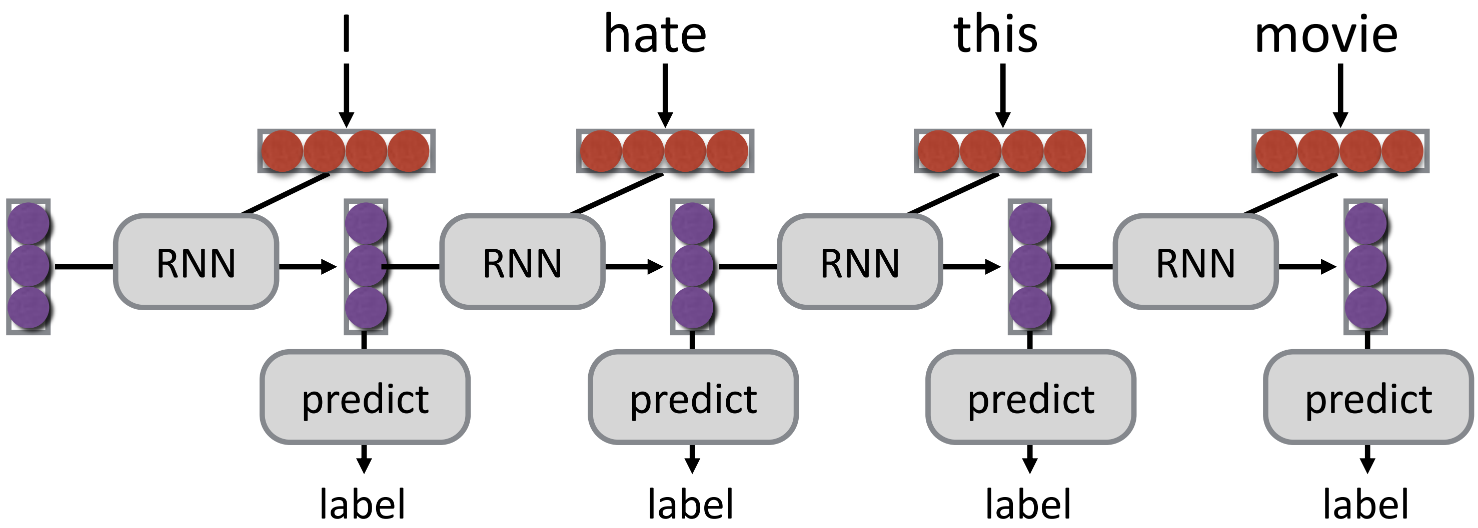
Recurrent Neural Networks

Tools to “remember” information

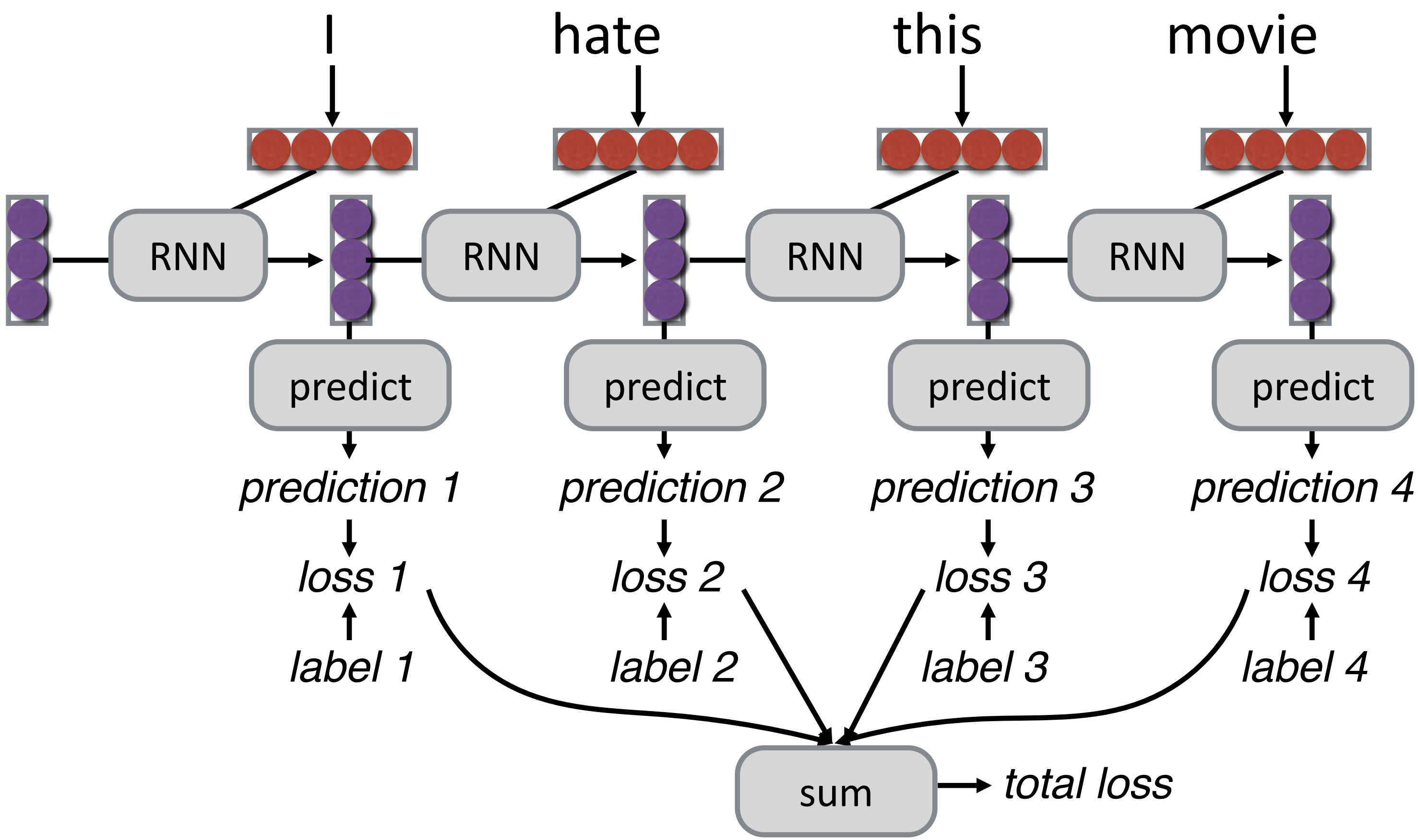


Unrolling in Time

What does processing a sequence look like?

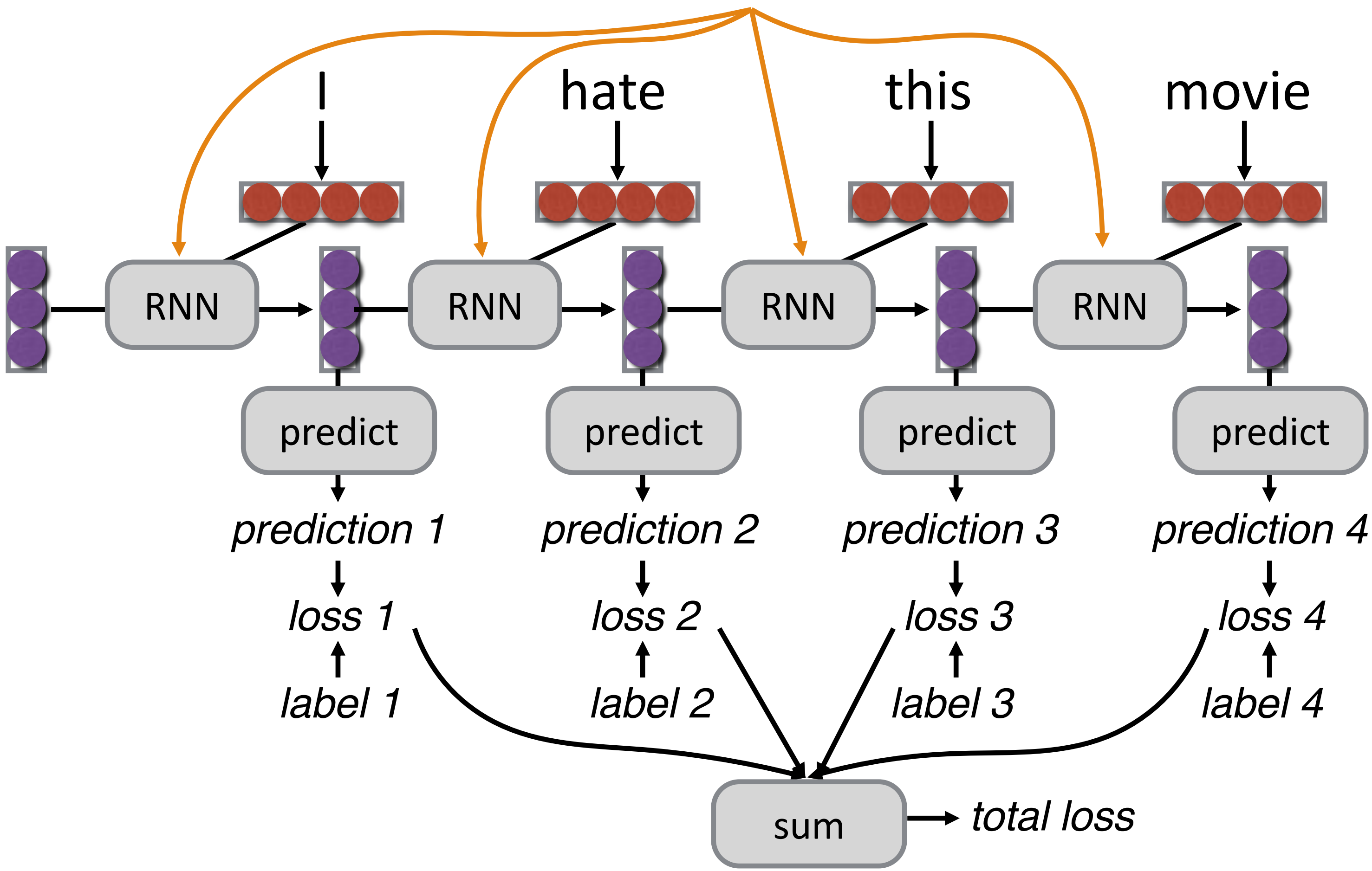


Training RNNs



Parameter Tying

Parameters are shared! Derivatives are accumulated.



What Can RNNs Do?

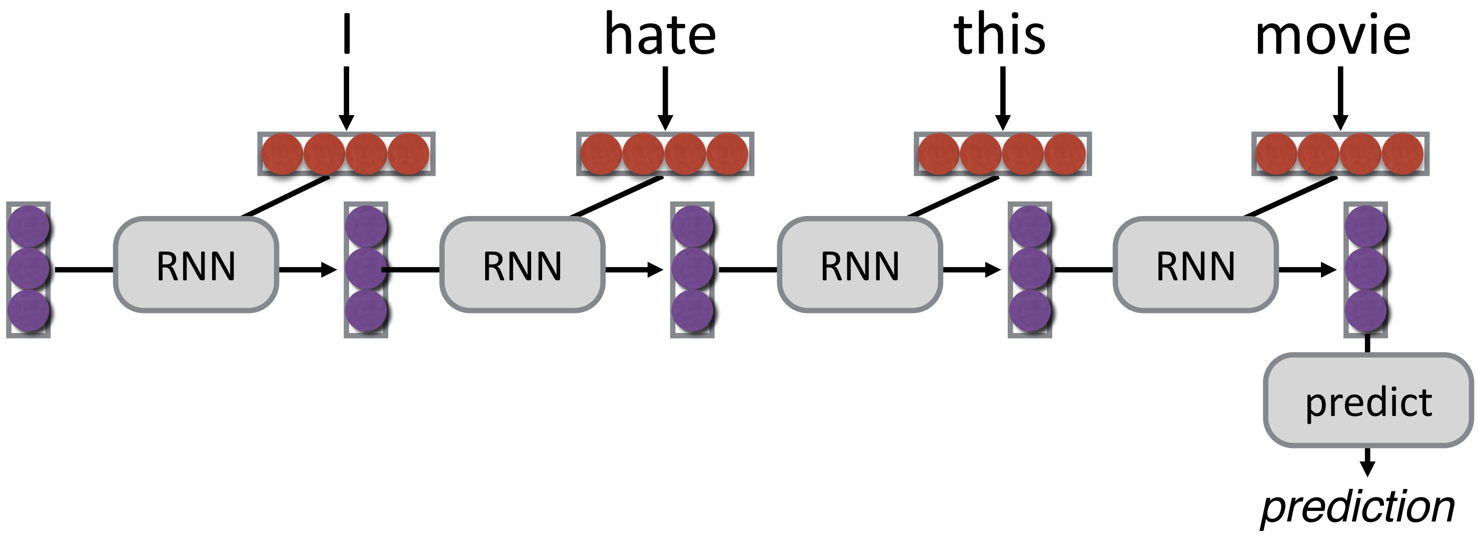
Represent a sentence

- Read whole sentence, make a prediction

Represent a context within a sentence

- Read context up until that point

Representing Sentences

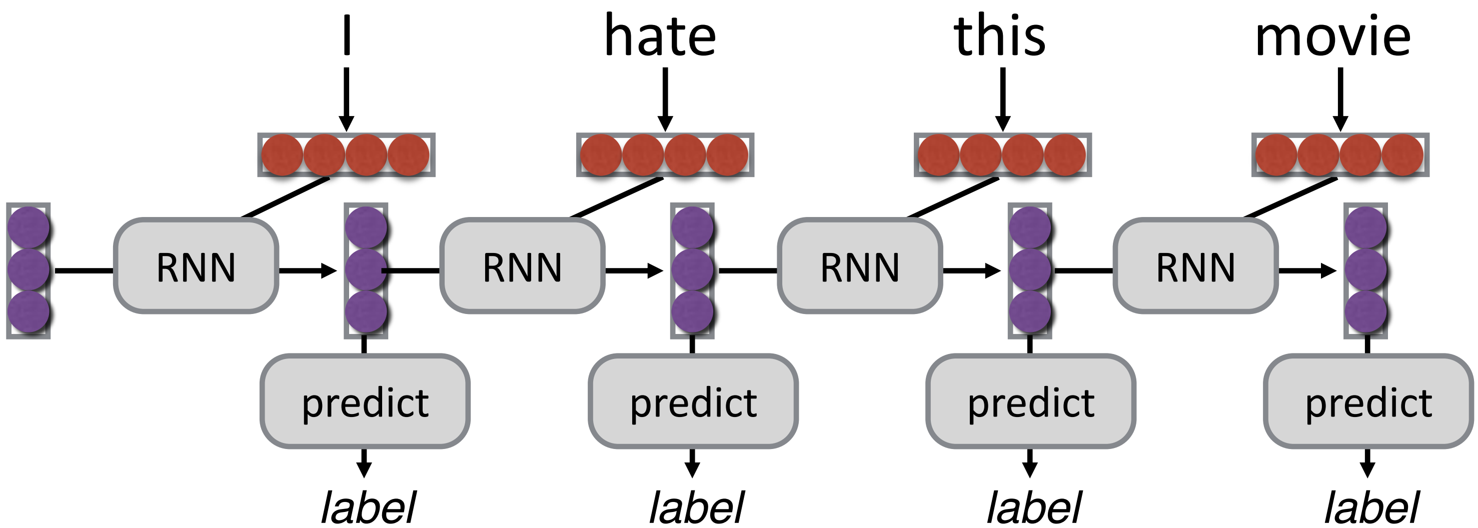


Sentence classification

Conditioned generation

Retrieval

Representing Contexts



Tagging

Language Modeling

Calculating Representations for Parsing, etc.

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.

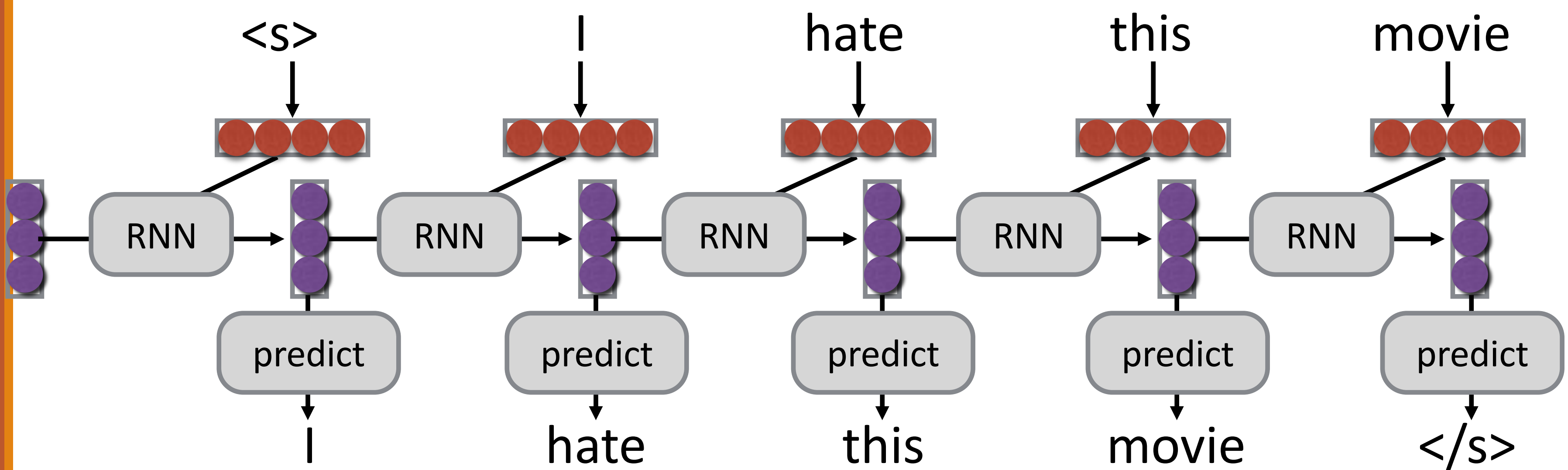
"I'm afraid I've definitely been suspended from power, no chance—indeed?" said Snape. He put his head back behind them and read groups as they crossed a corner and fluttered down onto their ink lamp, and picked up his spoon. The doorbell rang. It was a lot cleaner down in London.

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

Calculating the Probability of a Sentence

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

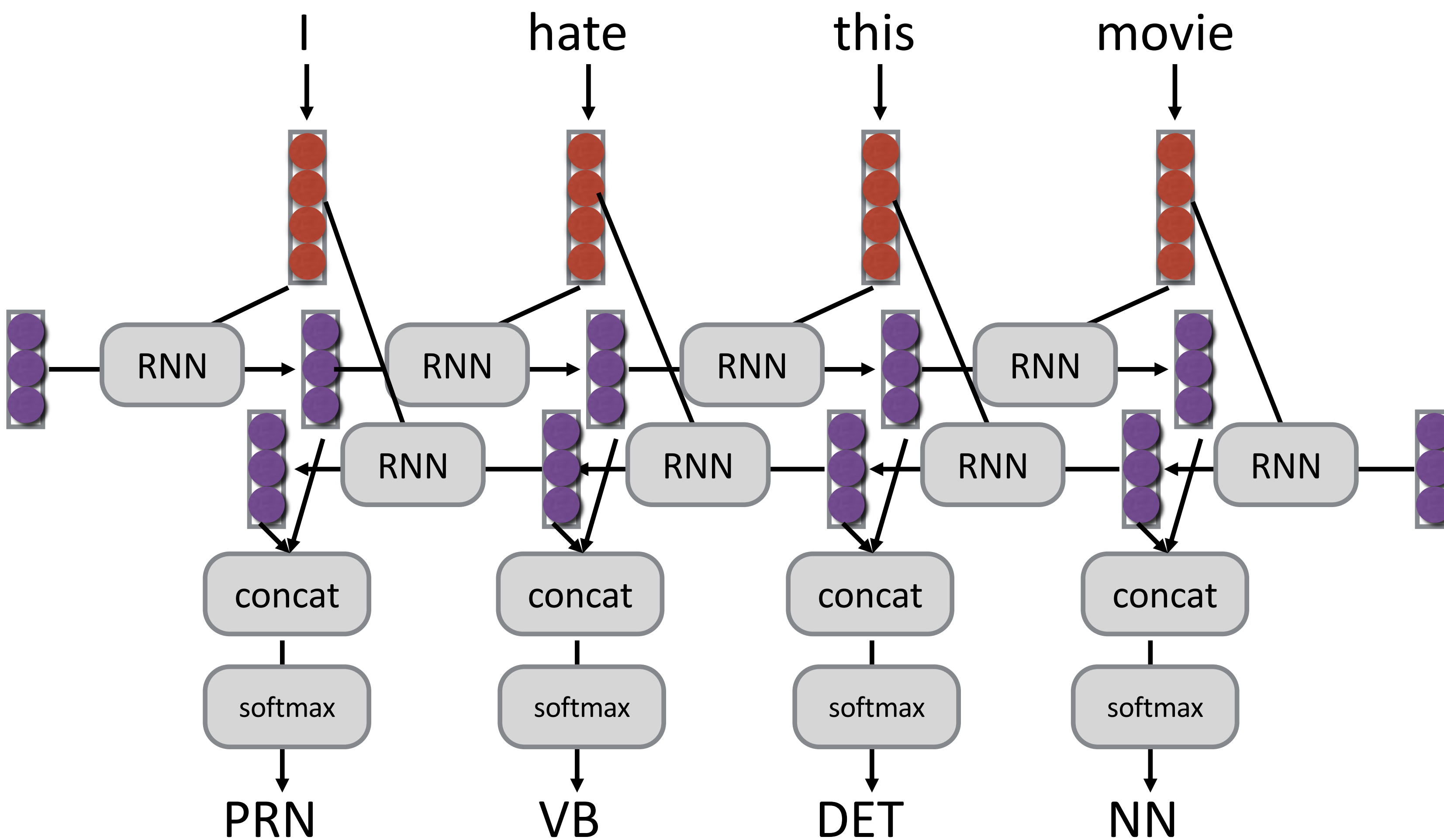
Language Models with RNNs



At each step, calculate probability of next word

Bi-directional RNNs

A simple extension, run the RNN in both directions



Conditional Language Modeling for Machine Translation

Conditional Language Models

Not just generate text, generate text according to some specification

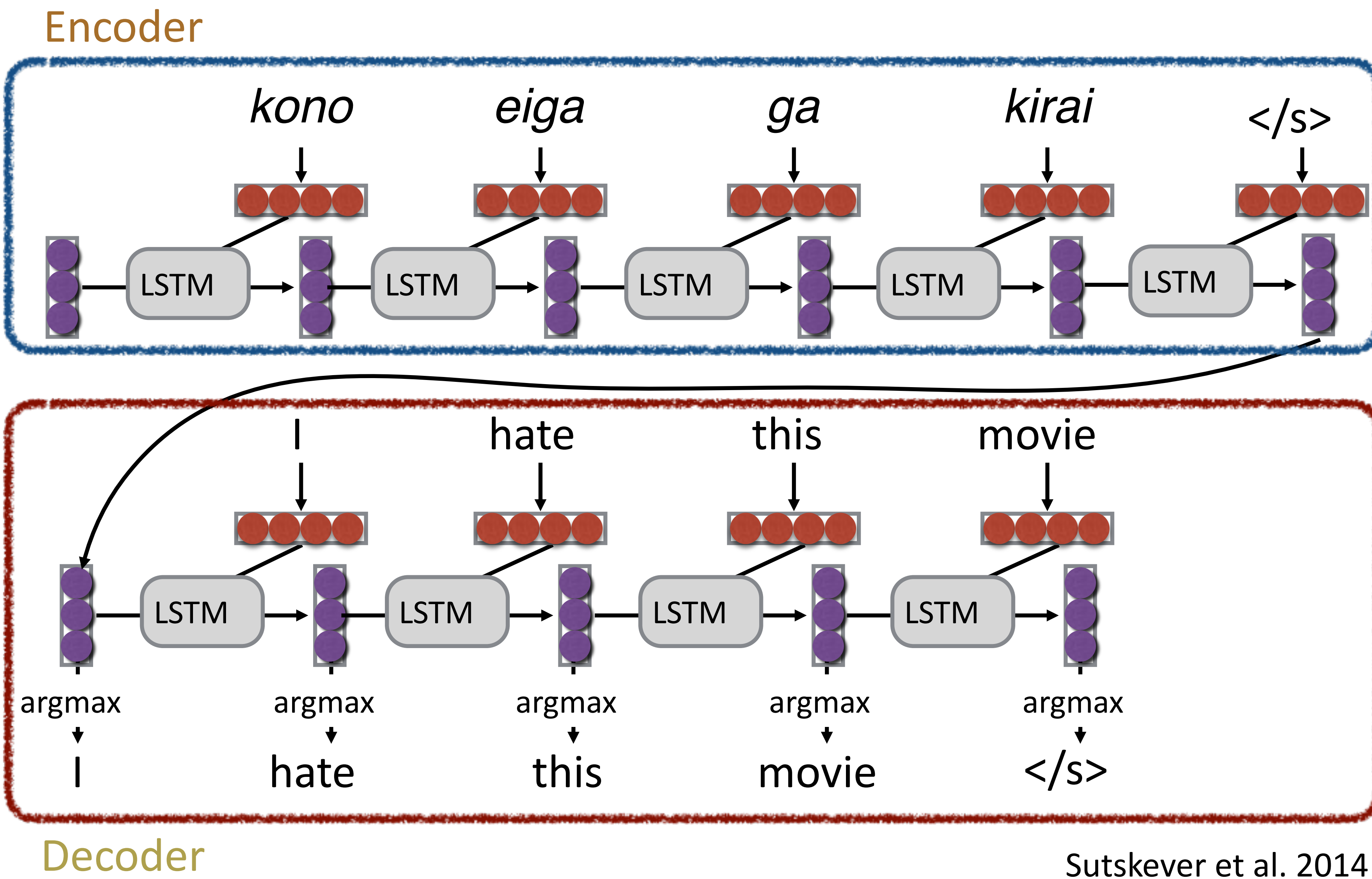
Input X	Output Y (**Text**)	Task
Structured Data	NL Description	NL Generation
English	Japanese	Translation
Document	Short Description	Summarization
Utterance	Response	Response Generation
Image	Text	Image Captioning
Speech	Transcript	Speech Recognition

Conditional Language Models

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

Added Context!

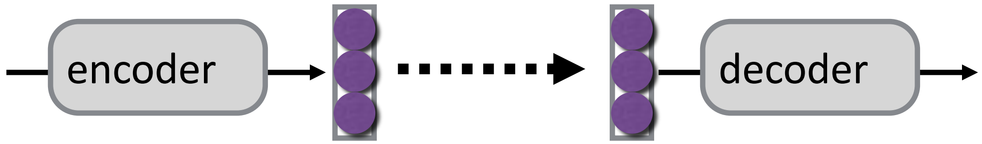
One type of Conditional LM



Sutskever et al. 2014

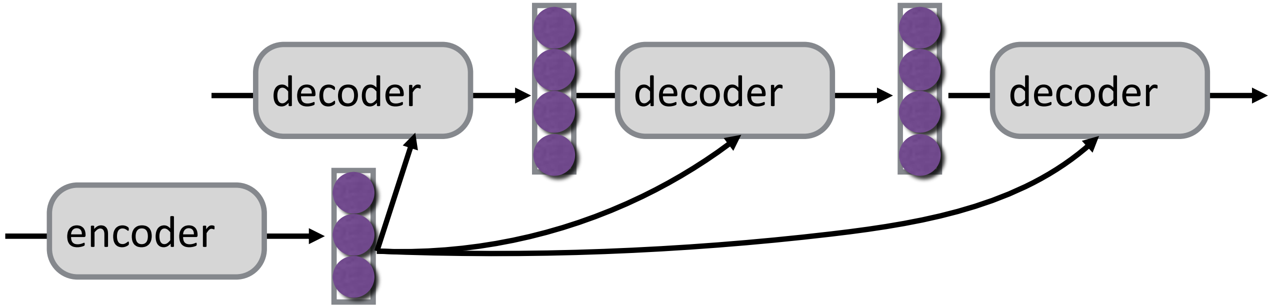
How to pass the hidden state?

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



Transform (can be different dimensions)

Input at every time step (Kalchbrenner & Blunsom 2013)



Training Conditional LMs

Get parallel corpus of inputs and outputs

Maximize likelihood

Standard corpora for MT:

- WMT Conference on Machine Translation runs an evaluation every year with large-scale (e.g. 10M sentence) datasets
- Smaller datasets, e.g. 200k sentence TED talks from IWSLT, can be more conducive to experimentation

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 ~ P(y_j | X, y_1, ..., 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

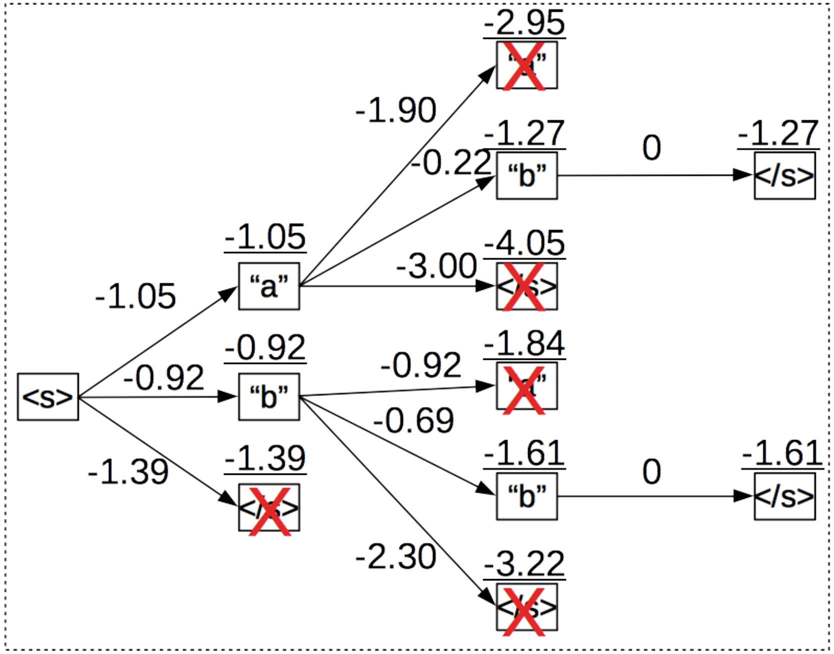
$$\begin{aligned}&\text{while } y_{j-1} \mathrel{!}= \text{“</s>”:}\\ &\quad y_j = \operatorname{argmax} P(y_j \mid X, y_1, \ldots, y_{j-1})\end{aligned}$$

Not exact, which causes real problems:

1. Will often generate the “easy” words first
2. Will prefer multiple common words to one rare word

Beam Search

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



How do we Evaluate the Quality of MT?

Evaluating MT Quality

Why do we want to do it?

- Want to rank systems
- Want to evaluate incremental changes
- What to make scientific claims

How not to do it

- “Back translation”
- The vodka is not good

Human Evaluation of MT v. Automatic Evaluation

Human evaluation is

- Ultimately what we're interested in, but
- Very time consuming
- Not re-usable

Automatic evaluation is

- Cheap and reusable, but
- Not necessarily reliable

Manual Evaluation

Source: Estos tejidos están analizados, transformados y congelados antes de ser almacenados en Hema-Québec, que gestiona también el único banco público de sangre del cordón umbilical en Quebec.

Reference: These tissues are analyzed, processed and frozen before being stored at Héma-Québec, which manages also the only bank of placental blood in Quebec.

Translation	Rank 1 Best	2	3	4	5 Worst
These weavings are analyzed, transformed and frozen before being stored in Hema-Quebec, that negotiates also the public only bank of blood of the umbilical cord in Quebec.	○	○	○	○	●
These tissues analysed, processed and before frozen of stored in Hema-Québec, which also operates the only public bank umbilical cord blood in Quebec.	○	○	●	○	○
These tissues are analyzed, processed and frozen before being stored in Hema-Québec, which also manages the only public bank umbilical cord blood in Quebec.	○	●	○	○	○
These tissues are analyzed, processed and frozen before being stored in Hema-Quebec, which also operates the only public bank of umbilical cord blood in Quebec.	●	○	○	○	○
These fabrics are analyzed, are transformed and are frozen before being stored in Hema-Québec, who manages also the only public bank of blood of the umbilical cord in Quebec.	○	○	○	●	○

Goals for Automatic Evaluation

No cost evaluation for incremental changes

Ability to rank systems

Ability to identify which sentences we're doing poorly on, and categorize errors

Correlation with human judgments

Interpretability of the score

Methodology

- Comparison against reference translations
- Intuition: closer we get to human translations, the better we're doing
- Could use WER like in speech recognition?

Word Error Rate

Levenshtein Distance (also known as "edit distance")

Minimum number of insertions, substitutions, and deletions needed to transform one string into another

Useful measure in speech recognition

- This shows how easy it is to recognize speech
- This shows how easy it is to wreck a nice beach

Problems with WER

Unlike speech recognition we don't have the assumption of **exact match against the reference or linearity**

In MT there can be many possible (and equally valid) ways of translating a sentence, and phrases can be rearranged.

1

Compare against lots of test sentences

2

Use multiple reference translations for each test sentence

3

Look for phrase / n-gram matches, allow movement

Solutions

BLEU

Bi**L**ingual **E**valuation **U**nderstudy

Uses multiple reference translations

Look for n-grams that occur anywhere in the sentence

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida.
Ref 2	Orejuela appeared calm while being escorted to the plane that would take him to Miami, Florida.
Ref 3	Orejuela appeared calm as he was being led to the American plane that was to carry him to Miami in Florida.
Ref 4	Orejuela seemed quite calm as he was being led to the American plane that would take him to Miami in Florida.

Multiple references

$$p_n = \frac{\sum_{S \in C} \sum_{ngram \in S} Count_{matched}(ngram)}{\sum_{S \in C} \sum_{ngram \in S} Count(ngram)}$$

n-gram precision

BLEU MODIFIES THIS PRECISION TO ELIMINATE REPETITIONS THAT OCCUR ACROSS SENTENCES.

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him **to Miami**, Florida.
Ref 2	Orejuela appeared calm while being escorted to the plane that would take him **to Miami**, Florida.
Ref 3	Orejuela appeared calm as he was being led to the American plane that was to carry him **to Miami** in Florida.
Ref 4	Orejuela seemed quite calm as he was being led to the American plane that would take him in Florida. **to Miami**

Multiple references

"to Miami" can only be counted as correct once

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida.
Ref 2	Orejuela appeared calm while being escorted to the plane that would take him to Miami, Florida.
Ref 3	Orejuela appeared calm as he was being led to the American plane that was to carry him to Miami in Florida.
Ref 4	Orejuela seemed quite calm as he was being led to the American plane that would take him to Miami in Florida.

Hyp	appeared calm when he was taken to the American plane, which will to Miami, Florida.

American, Florida, Miami, Orejuela, **appeared**, as, being, calm, carry, escorted, he, him, in, led, **plane**, quite, seemed, take, that, the, **to**, **to**, to, **was** , was, **which**, while, **will**, would, **,, .**

1-gram precision = 15/18

Hyp	**appeared calm** when **he was** taken **to the American plane** , **which will to Miami , Florida .**

American plane, **Florida .**, **Miami ,**, Miami in, Orejuela appeared, Orejuela seemed, **appeared calm**, as he, being escorted, being led, calm as, calm while, carry him, escorted to, **he was**, him to, in Florida, led to, plane that, plane which, quite calm, seemed quite, take him, that was, that would, **the American**, the plane, **to Miami**, to carry, **to the**, was being, was led, was to, **which will**, while being, will take, would take, , Florida

2-gram precision = 10/17

Hyp	**appeared calm** when **he was** taken **to the American plane** , **which will to Miami , Florida .**

n-gram precision

Hyp	appeared calm when he was taken to the American plane, which will to Miami, Florida.

1-gram precision = 15/18 = .83

2-gram precision = 10/17 = .59

3-gram precision = 5/16 = .31

4-gram precision = 3/15 = .20

- Geometric average

(0.83 * 0.59 * 0.31 * 0.2)^(1/4) = 0.417

or equivalently

exp(ln .83 + ln .59 + ln .31 + ln .2/4) = 0.417

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida.
Ref 2	Orejuela appeared calm while being escorted to the plane that would take him to Miami, Florida.
Ref 3	Orejuela appeared calm as he was being led to the American plane that was to carry him to Miami in Florida.
Ref 4	Orejuela seemed quite calm as he was being led to the American plane that would take him to Miami in Florida.

Hyp	to the American plane

Is this better?

Hyp	to the American plane

1-gram precision = 4/4 = 1.0

2-gram precision = 3/3 = 1.0

3-gram precision = 2/2 = 1.0

4-gram precision = 1/1 = 1.0

$$\exp(\ln 1 + \ln 1 + \ln 1 + \ln 1) = 1$$

Brevity Penalty

c is the length of the corpus of hypothesis translations

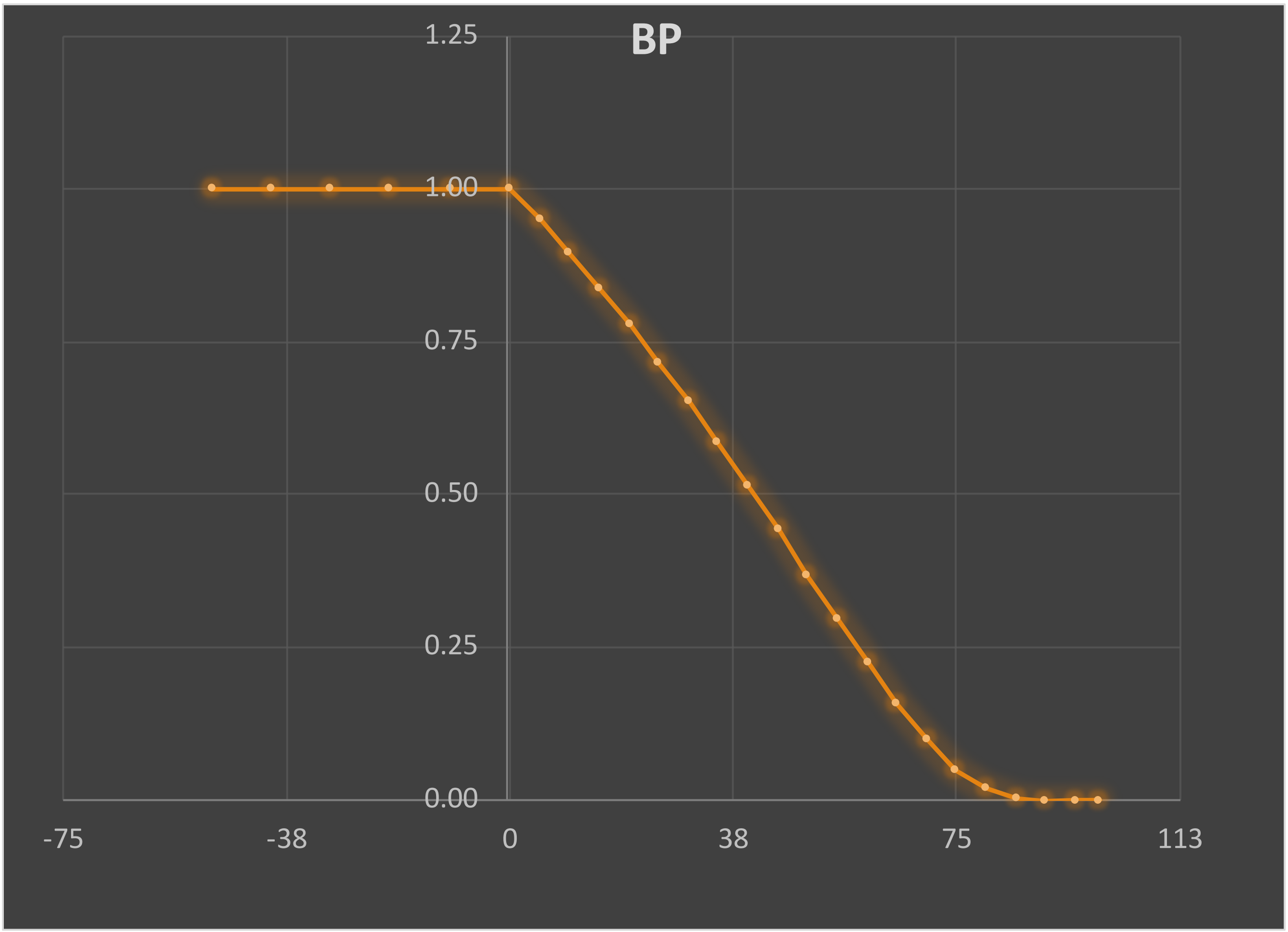
r is the effective reference corpus length

The effective reference corpus length is the sum of the single reference translation from each set that is closest to the hypothesis translation.

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

Brevity Penalty

MT is Longer MT is Shorter



Difference with effective reference length (%)

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida. $r = 20$
Hyp	appeared calm when he was taken to the American plane, which will to Miami, Florida. $c = 18$

$$BP = \exp(1-(20/18)) = 0.89$$

Ref 1	Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida. $r = 20$
Hyp	to the American plane $c = 4$

$$BP = \exp(1-(20/4)) = 0.02$$

BLEU

Geometric average of the n-gram precisions

Optionally weight them with w

Multiplied by the brevity penalty

$$\text{Bleu} = \text{BP} * \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

BLEU

Hyp	appeared calm when he was taken to the American plane, which will to Miami, Florida.

$$\exp(1-(20/18)) * \exp((\ln .83 + \ln .59 + \ln .31 + \ln .2)/4) = \mathbf{0.374}$$

Hyp	to the American plane

$$\exp(1-(20/4)) * \exp((\ln 1 + \ln 1 + \ln 1 + \ln 1)/4) = \mathbf{0.018}$$

Problems with BLEU

Synonyms and paraphrases are only handled if they are in the set of multiple reference translations

The scores for **words are equally weighted** so missing out on content-bearing material brings no additional penalty.

The brevity penalty is a stop-gap measure to compensate for the fairly serious problem of not being able to calculate **recall**.

WER - word error rate

PI-WER - position independent WER

METEOR - **M**etric for **E**valuation of **T**ranslation with **E**xplicit **OR**dering

TERp - **T**ranslation **E**dit **R**ate **p**lus

More Metrics

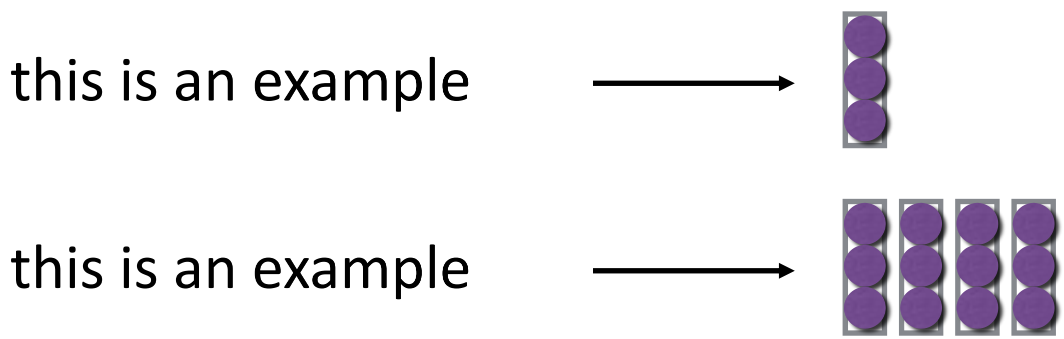
Attention

Sentence Representations

You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!

Sentence Representations

But what if we could use multiple vectors, based on the length of the sentence?



Basic Idea

Encode each word in the sentence into a vector

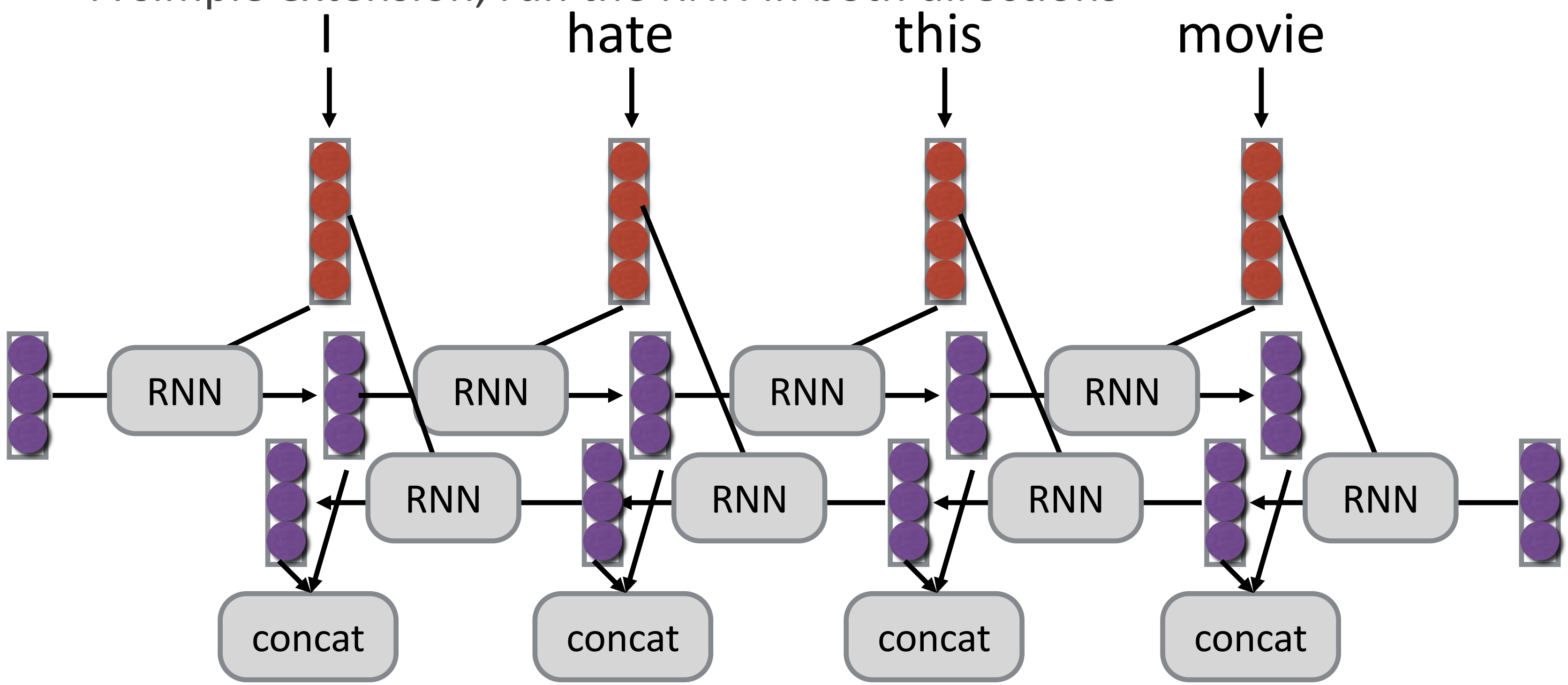
When decoding, perform a linear combination of these vectors, weighted by “attention weights”

Use this combination in picking the next word

Neural Machine Translation by Jointly Learning to Align and Translate
by Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 2015

Encoder Bi-RNNs

A simple extension, run the RNN in both directions

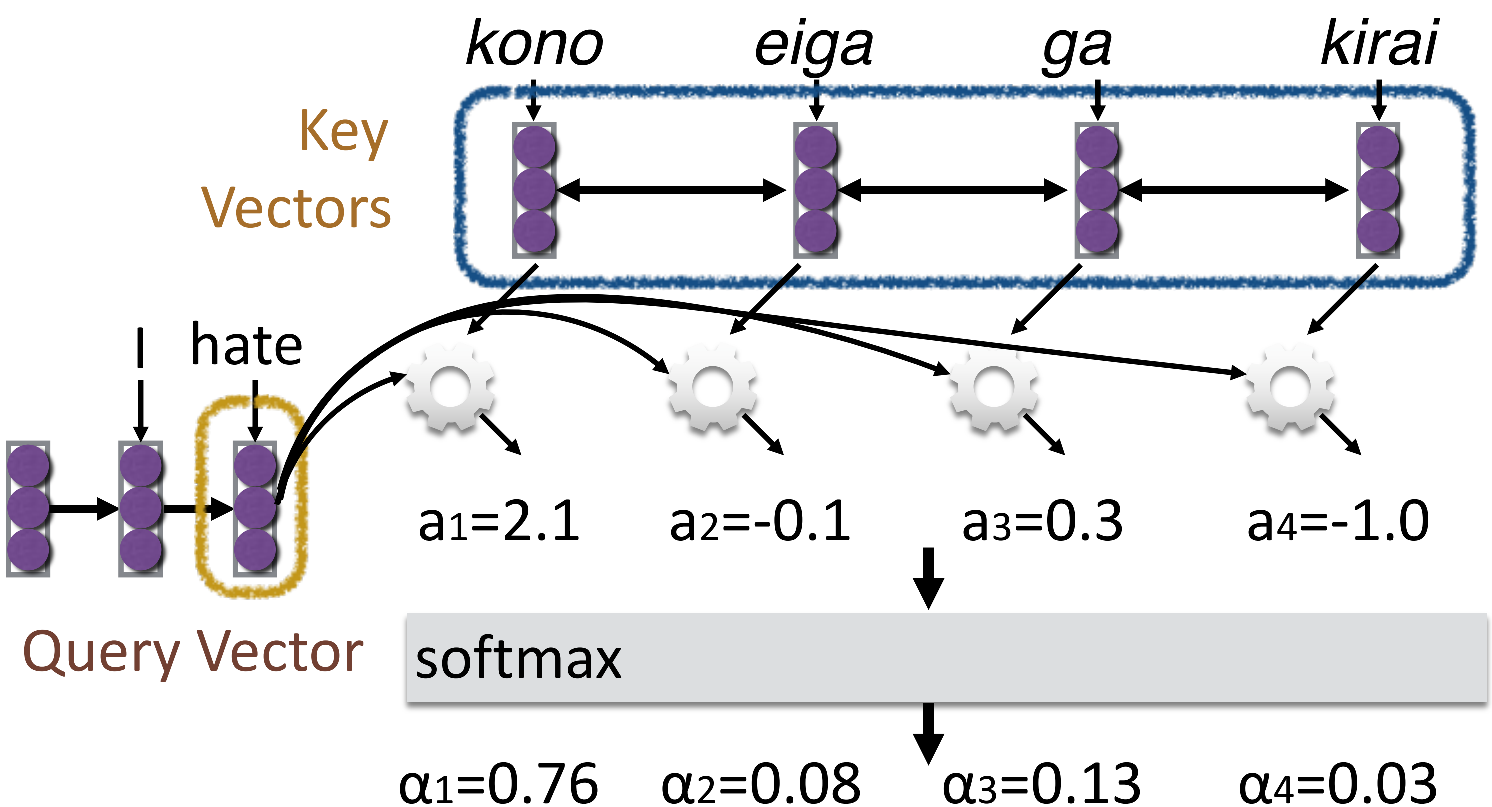


Calculating Attention (1)

Use "query" vector (decoder state) and "key" vectors (all encoder states)

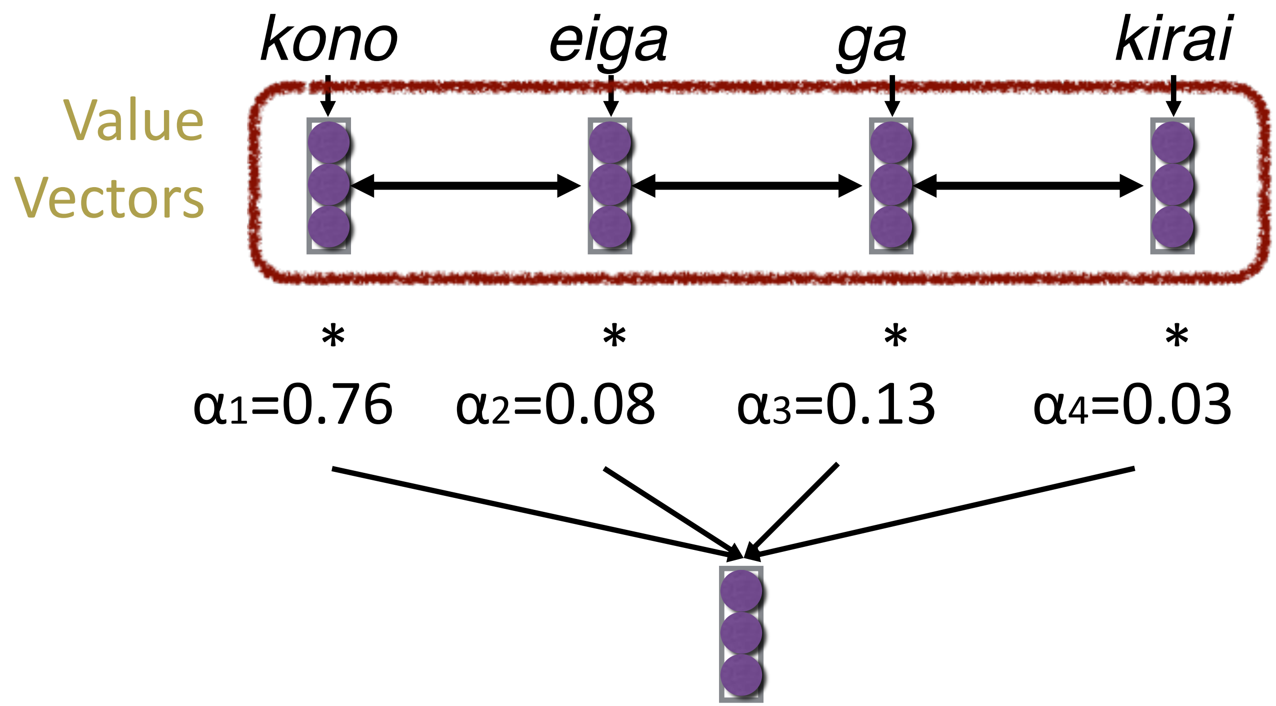
For each query-key pair, calculate weight

Normalize to add to one using softmax



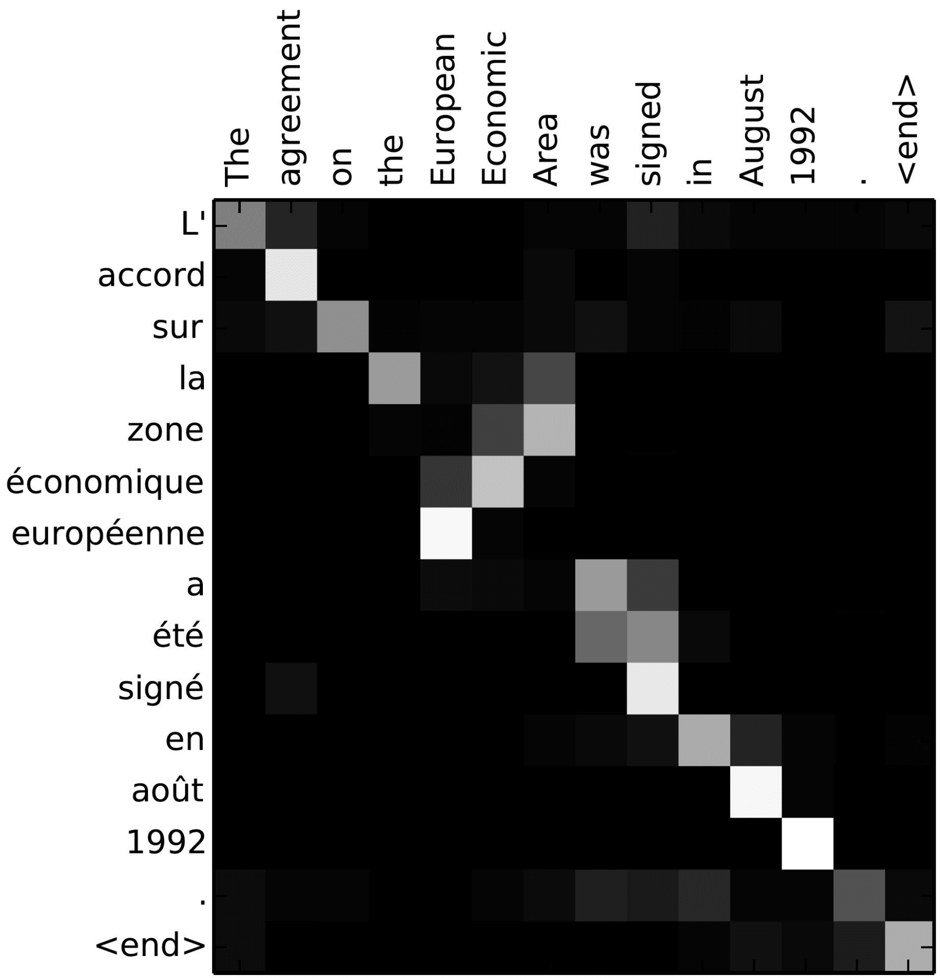
Calculating Attention (2)

Combine together value vectors (usually encoder states, like key vectors) by taking the weighted sum



Use this in any part of the model you like

A Graphical Example



Attention Score Functions (1)

$\boldsymbol{q}$ is the query and $\boldsymbol{k}$ is the key

Multi-layer Perceptron (Bahdanau et al. 2015)

$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{w}_2^{\intercal} \tanh(W_1[\boldsymbol{q}; \boldsymbol{k}])$$

- Flexible, often very good with large data

Bilinear (Luong et al. 2015)

$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{q}^{\intercal} W \boldsymbol{k}$$

Attention Score Functions (2)

Dot Product (Luong et al. 2015)

$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{q}^\intercal \boldsymbol{k}$$

- No parameters! But requires sizes to be the same.

Scaled Dot Product (Vaswani et al. 2017)

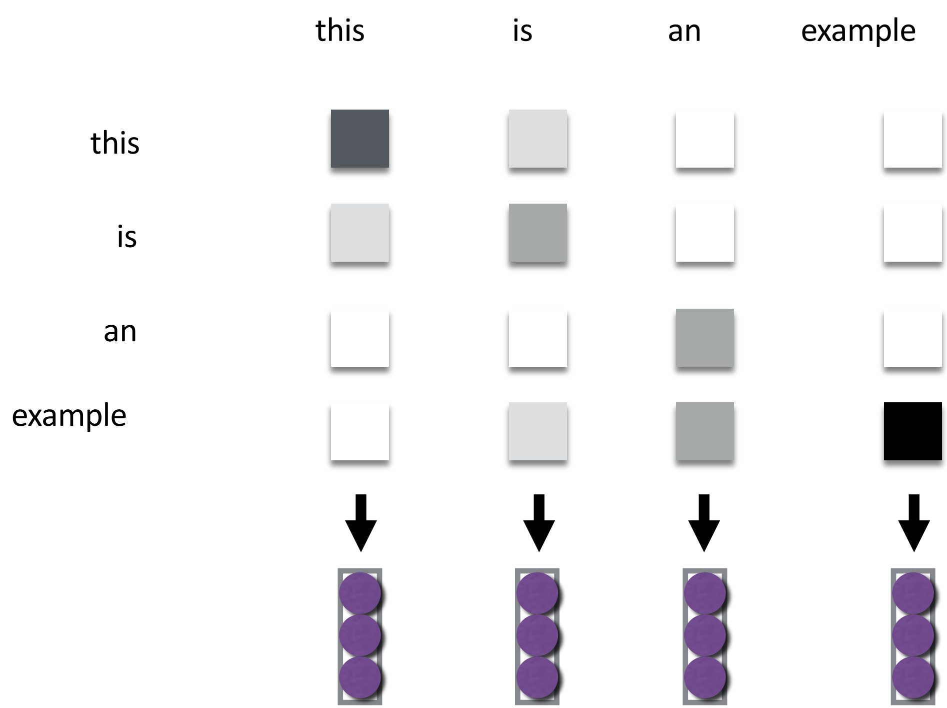
- Problem: scale of dot product increases as dimensions get larger
- Fix: scale by size of the vector

$$a(\boldsymbol{q}, \boldsymbol{k}) = \frac{\boldsymbol{q}^\intercal \boldsymbol{k}}{\sqrt{|\boldsymbol{k}|}}$$

Extensions to Attention

Intra-Attention / Self Attention
(Cheng et al. 2016)

Each element in the sentence attends to other elements → context sensitive encodings!



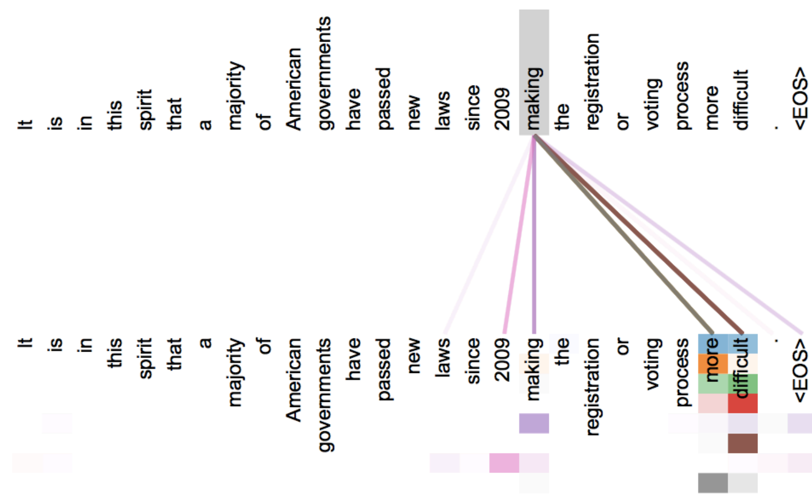
Multi-headed Attention

Idea: multiple attention "heads" focus on different parts of the sentence

- e.g. Different heads for "copy" vs regular (Allamanis et al. 2016)

Target			Attention Vectors	λ
m_1	set	$\boldsymbol{\alpha} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	0.012
		$\boldsymbol{\kappa} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	
m_2	use	$\boldsymbol{\alpha} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	0.974
		$\boldsymbol{\kappa} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	
m_3	browser	$\boldsymbol{\alpha} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	0.969
		$\boldsymbol{\kappa} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	
m_4	cache	$\boldsymbol{\alpha} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	0.583
		$\boldsymbol{\kappa} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	
m_5	END	$\boldsymbol{\alpha} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	0.066
		$\boldsymbol{\kappa} =$	<s> { this . use Browser Cache = use Browser Cache ; } </s>	

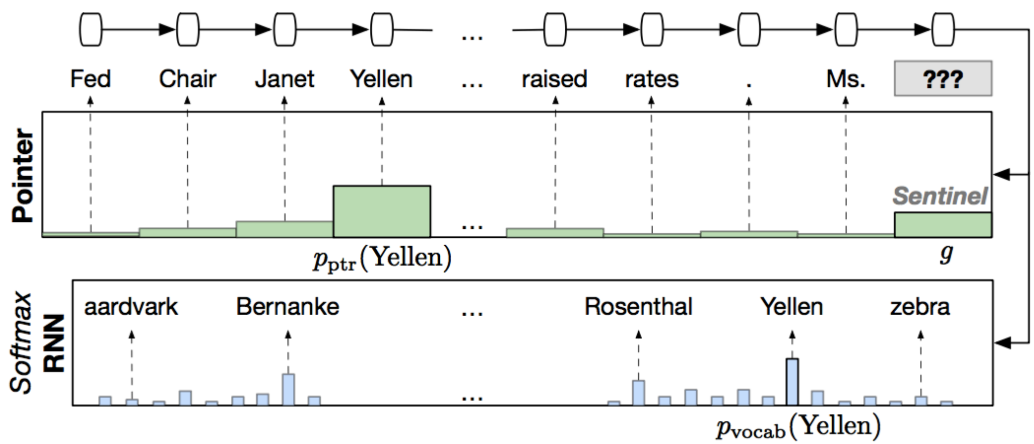
- Or multiple independently learned heads (Vaswani et al. 2017)



- Or one head for every hidden node! (Choi et al. 2018)

Attending to Previously Generated Things

In language modeling, attend to the previous words (Merity et al. 2016)



$$p(\mathrm{Yellen}) = g\, p_{\mathrm{vocab}}(\mathrm{Yellen}) + (1-g)\, p_{\mathrm{ptr}}(\mathrm{Yellen})$$

In translation, attend to either input or previous output (Vaswani et al. 2017)

An Interesting Case Study: “Attention is All You Need”

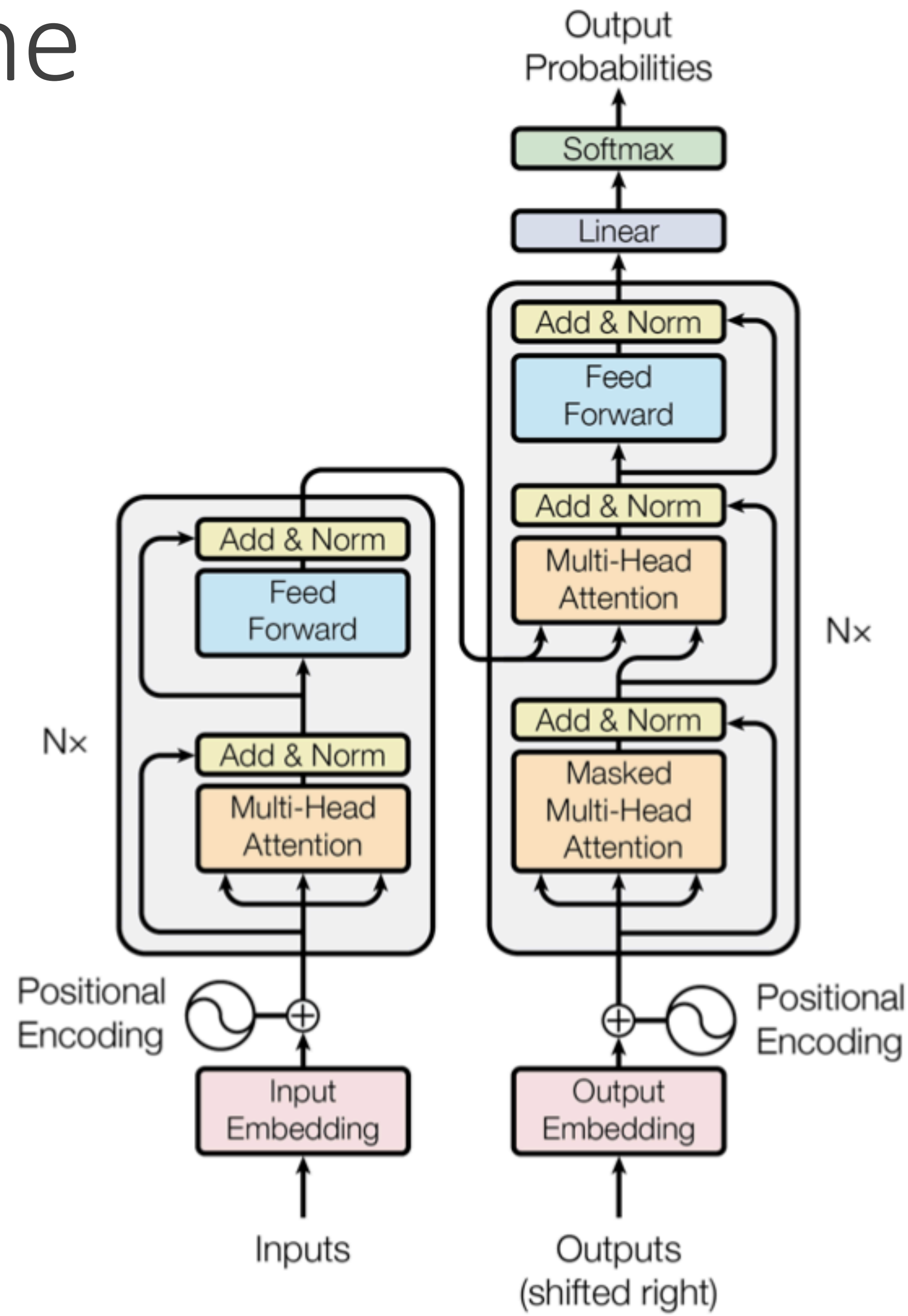
(Vaswani et al. 2017)

Summary of the “Transformer" (Vaswani et al. 2017)

A sequence-to-sequence model based entirely on attention

Also have attention on the output side! Calculate probability of next word by attention over previous words.

Fast: only matrix multiplications



Attention Tricks

Self Attention: Each layer combines words with others

Multi-headed Attention: 8 attention heads function independently

Normalized Dot-product Attention: Remove bias in dot product when using large networks

Positional Encodings: Make sure that even if we don't have RNN, can still distinguish positions

Training Tricks

Layer Normalization: Help ensure that layers remain in reasonable range

Specialized Training Schedule: Adjust default learning rate of the Adam optimizer

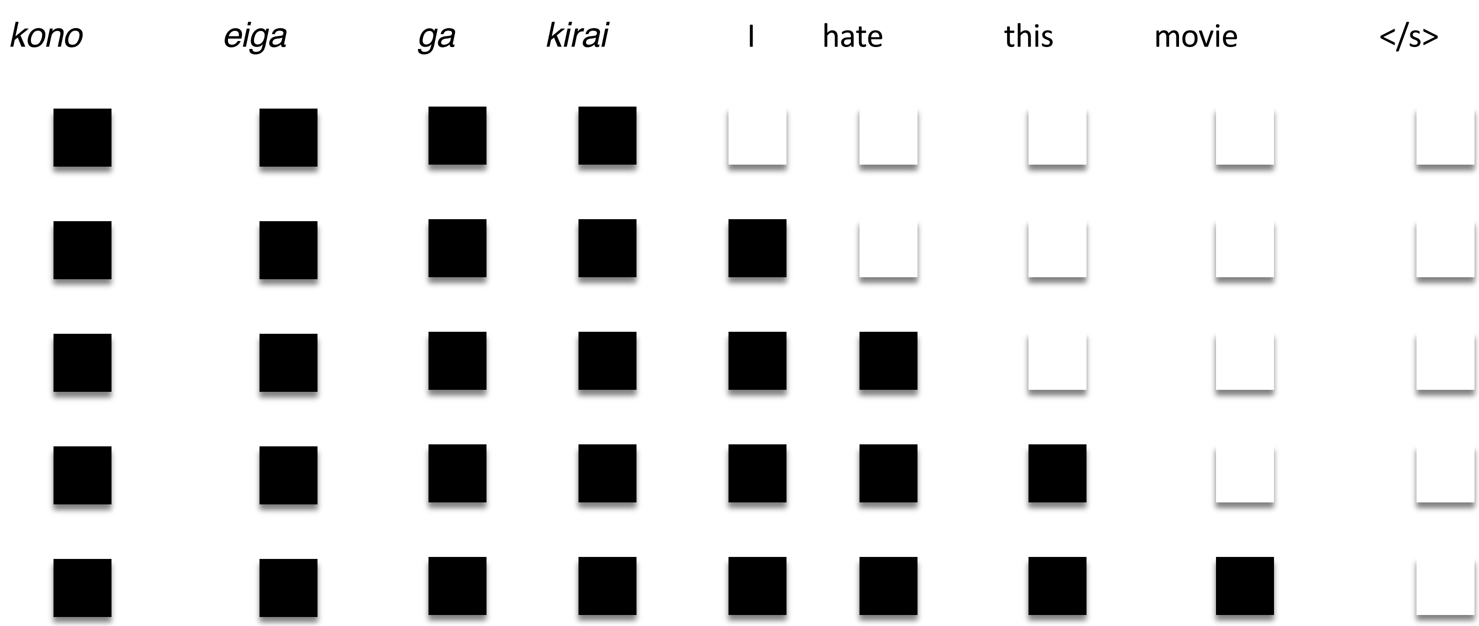
Label Smoothing: Insert some uncertainty in the training process

Masking for Efficient Training

Masking for Training

We want to perform training in as few operations as possible using big matrix multiplies

We can do so by “masking” the results for the output



How to Get Started?

Getting Started

Find training data, (e.g. TED talks from IWSLT), in your favorite language

Download a toolkit (e.g. OpenNMT, fairseq, Sockeye, xnmt) and run it on the data

Calculate the BLEU score and look at the results

Think of what's going right, what's going wrong!

Questions?

To Learn More:
"Neural Machine Translation and Sequence-to-sequence Models: A Tutorial"