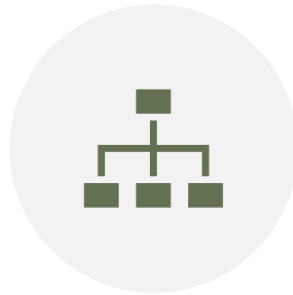


# Reminders



QUIZ ON CHAPTERS 15&16  
IS DUE TONIGHT BY  
11:59PM.



HOMEWORK 9 OR PROJECT  
MILESTONE 1 ARE DUE  
WEDNESDAY



I DISTRIBUTED DATA FOR  
HW9 FOR EVERYONE WHO  
COMPLETED IRB TRAINING

# Semantic Role Labeling

JURAFSKY AND MARTIN CHAPTER 20

# Events and their Participants

A **purchasing** event and its participants can be described by a wide variety of surface forms.

1. XYZ corporation bought the stock.
2. They sold the stock to XYZ corporation.
3. The stock was bought by XYZ corporation.
4. The purchase of the stock by XYZ corporation...
5. The stock purchase by XYZ corporation...

Commonality: there was a **purchase** event, the participants were **XYZ Corp** and some amount of **stock**, and **XYZ Corp** was **the buyer**.

**Semantic Role Labels** give a shallow semantic representation of the event and its arguments.

# Semantic Roles

Last time we discussed *neo-Davidsonian* event representations.

*Sasha broke the window*

$\exists e, x, y \text{ Breaking}(e) \wedge \text{Breaker}(e, \text{Sasha}) \wedge \text{BrokenThing}(e, y) \wedge \text{Window}(y)$

*Pat opened the door.*

$\exists e, x, y \text{ Opening}(e) \wedge \text{Opener}(e, \text{Pat}) \wedge \text{OpenedThing}(e, y) \wedge \text{Door}(y)$

The **semantic role** of the subject of the *break* is **Breaker**

The semantic of the subject of the *open* is **Opener**

These **deep roles** are specific to each event.

# Thematic Roles

*Breakers* and *Openers* have something in common. They are both volitional actors, usually animate, and they have a direct causal responsibility for their events.

**Thematic roles** are a way to capture this semantic commonality between these roles. In this case, both *Breakers* and *Openers* fill the thematic role of AGENT.

AGENT is the thematic role that represents an abstract idea such as *volitional causation*.

BrokenThing and OpenedThing, are both prototypically inanimate objects that are affected in some way by the action. The semantic role for participant most directly affected by an event is THEME ;

# Thematic Roles

Thematic Role	Definition
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

# Thematic Roles

Thematic Role	Definition
AGENT	<i>The player</i> kicked the ball.
EXPERIENCER	<i>Dan</i> has a cough and a fever.
FORCE	<i>The coronavirus</i> spread rapidly through the country
THEME	The wind blows <i>debris</i> from the street into our yard
RESULT	The city implemented <i>a stay-at-home policy for non-essential personnel</i>
CONTENT	Chris asked “ <i>You met Rebecca at a supermarket?</i> ”
INSTRUMENT	He poached catfish, stunning them with <i>a shocking device...</i>
BENEFICIARY	Joe makes hotel reservations for <i>his boss</i> .
SOURCE	I flew in <i>from Boston</i> .
GOAL	I drove <i>to Portland</i> .

# Verb Alternation

Semantic roles act as a shallow meaning representation that allow systems make simple inferences that aren't possible from the surface string of words, or from the parse tree.

Arguments for verbs alternate in their positions, which makes pure surface analysis difficult.

*John broke the window.*

AGENT            THEME

*John broke the window with a rock.*

AGENT            THEME            INSTRUMENT

*The rock broke the window.*

INSTRUMENT            THEME

*The window broke.*

THEME

*The window was broken by John.*

THEME                            AGENT



# Problems with Thematic Roles

Analysis of thematic roles should be useful for handling verb alternation. However, there is no single, standard set of thematic roles.

And it's quite difficult to come up with a formal definition for things like AGENT, THEME, or INSTRUMENT.

For example, there are two kinds of INSTRUMENTS, intermediary instruments that can appear as subjects and enabling instruments that cannot:

1. The cook opened the jar with the new gadget.  
The new gadget opened the jar.
2. Shelly ate the sliced banana with a fork.  
\*The fork ate the sliced banana.

Different theories of thematic roles treat these differently, which causes fragmentation across theories.

# Generalized Semantic Roles

Instead of creating a more fine-grained inventory of Thematic Roles, research in NLP has shifted in the direction of coarser roles.

You may see terms like PROTO-AGENT and PROTO-PATIENT, which are generalized roles that express roughly agent-like and roughly patient-like meanings. These meanings are defined a set of heuristics.

A second direction that NLP goes in is to define semantic roles that are specific to each verb, or to a group of semantically related verbs or nouns.

Lexical resources that make use of this second direction are **PropBank** and **FrameNet**.

# PropBank

PropBank, is a resource of Penn TreeBank sentences annotated with semantic roles. It was created by Martha Palmer at UPenn.

Because defining universal thematic roles is difficult, PropBank defines semantic roles for each verb sense.

Each verb has a specific set of roles, given by numbers: Arg0, Arg1, Arg2.

In general, Arg0 represents the PROTO-AGENT, and Arg1, the PROTO-PATIENT.



Martha Palmer

# PropBank Frame File

## Agree.01

Arg0: Agreeer

Arg1: Proposition

Arg2: Other entity agreeing

← Glosses to be read by humans

Example 1:

[Arg0 The group] *agreed* [Arg1 it wouldn't make an offer].

Example 2: [ArgM-TMP Usually] [Arg0 Chris] *agrees* [Arg2 with Ellie] [Arg1 on everything].

# PropBank Frame File

Increase.01 “go up incrementally”

Arg0: Causer of increase

Arg1: Thing increasing

Arg2: Amount increased by

Arg3: Start point

Arg4: End point

Example 1: [Arg0 Big Fruit Co. ] **increased** [Arg1 the price of bananas].

Example 2: [Arg1 The price of bananas] was **increased** again [Arg0 by Big Fruit Co. ]

Example 3: [Arg1 The price of bananas] **increased** [Arg2 5% ].

# ArgMs

PropBank also has a number of non-numbered arguments called ArgMs, which represent modification meanings. These are stable across predicates, so aren't listed with each frame file

<b>TMP</b>	when?	yesterday evening, now
<b>LOC</b>	where?	at the museum, in San Francisco
<b>DIR</b>	where to/from?	down, to Bangkok
<b>MNR</b>	how?	clearly, with much enthusiasm
<b>PRP/CAU</b>	why?	because ... , in response to the ruling
<b>REC</b>		themselves, each other
<b>ADV</b>	miscellaneous	
<b>PRD</b>	secondary predication	...ate the meat raw

# FrameNet

In order to make semantic inferences about price increase events, we want to make the connection across many different verbs, not just the verb **increase**.

Example 1: [Arg1 The price of bananas] **increased** [Arg2 5% ].

Example 2: [Arg1 The price of bananas] **rose** [Arg2 5% ].

Example 3: There has been a [Arg2 5% ] **rise** in [Arg1 the price of bananas]

FrameNet is another Semantic Role Labeling project that attempts to address just these kinds of problems.

PropBank labels roles specific to an individual verb, and FrameNet labels roles are specific to a **frame**.

# Frames

What is a frame? Consider the following set of words:

*reservation, flight, travel, buy, price, cost, fare, rates, plane*

They form coherent chunk of common-sense background information concerning air travel. The background knowledge that unites these words a **frame**.

The idea that groups of words are defined with respect to some background information is widespread in AI and cognitive science. Similar to the notion of a **script** that we saw before.



# Frame Elements

<b>Core Roles</b>	
ATTRIBUTE	The ATTRIBUTE is a scalar property that the ITEM possesses.
DIFFERENCE	The distance by which an ITEM changes its position on the scale.
FINAL_STATE	A description that presents the ITEM's state after the change in the ATTRIBUTE's value as an independent predication.
FINAL_VALUE	The position on the scale where the ITEM ends up.
INITIAL_STATE	A description that presents the ITEM's state before the change in the ATTRIBUTE's value as an independent predication.
INITIAL_VALUE	The initial position on the scale from which the ITEM moves away.
ITEM	The entity that has a position on the scale.
VALUE_RANGE	A portion of the scale, typically identified by its end points, along which the values of the ATTRIBUTE fluctuate.
<b>Some Non-Core Roles</b>	
DURATION	The length of time over which the change takes place.
SPEED	The rate of change of the VALUE.
GROUP	The GROUP in which an ITEM changes the value of an ATTRIBUTE in a specified way.

Frame: **Change position on a scale**

# Lexical Units

<b>VERBS:</b>	dwindle	move	soar	escalation	shift
advance	edge	mushroom	swell	explosion	tumble
climb	explode	plummet	swing	fall	
decline	fall	reach	triple	fluctuation	<b>ADVERBS:</b>
decrease	fluctuate	rise	tumble	gain	increasingly
diminish	gain	rocket		growth	
dip	grow	shift	<b>NOUNS:</b>	hike	
double	increase	skyrocket	decline	increase	
drop	jump	slide	decrease	rise	

Lexical Units that trigger the **change position on a scale** frame

# FrameNet+: Fast Paraphrastic Tripling of FrameNet

Ellie Pavlick<sup>1</sup> Travis Wolfe<sup>2,3</sup> Pushpendre Rastogi<sup>2</sup>

Chris Callison-Burch<sup>1</sup> Mark Dredze<sup>2,3</sup> Benjamin Van Durme<sup>2,3</sup>

<sup>1</sup>Computer and Information Science Department, University of Pennsylvania

<sup>2</sup>Center for Language and Speech Processing, Johns Hopkins University

<sup>3</sup>Human Language Technology Center of Excellence, Johns Hopkins University

## Abstract

We increase the lexical coverage of FrameNet through automatic paraphrasing. We use crowdsourcing to manually filter out bad paraphrases in order to ensure a high-precision resource. Our expanded FrameNet contains an additional 22K lexical units, a 3-fold increase over the current FrameNet, and achieves 40% better coverage when evaluated in a practical setting on New York Times data.

## 1 Introduction

*Frame semantics* describes a word in relation to real-world events, entities, and activities. Frame semantic analysis can improve natural language understanding (Fillmore and Baker, 2001), and

accurate, ambiguous, <b>apparent</b> , <b>apparently</b> , audible, axiomatic, <b>blatant</b> , <b>blatantly</b> , <b>blurred</b> , <b>blurry</b> , <b>certainly</b> , <b>clarify</b> , clarity, clear, clearly, <b>confused</b> , <b>confusing</b> , conspicuous, crystal-clear, <b>dark</b> , <b>definite</b> , <b>definitely</b> , <b>demonstrably</b> , <b>discernible</b> , <b>distinct</b> , evident, <b>evidently</b> , <b>explicit</b> , <b>explicitly</b> , <b>flagrant</b> , <b>fuzzy</b> , <b>glaring</b> , <b>imprecise</b> , <b>inaccurate</b> , <b>lucid</b> , manifest, <b>manifestly</b> , <b>markedly</b> , <b>naturally</b> , <b>notable</b> , <b>noticeable</b> , <b>obscure</b> , <b>observable</b> , obvious, obviously, <b>opaque</b> , <b>openly</b> , <b>overt</b> , <b>patently</b> , <b>perceptible</b> , <b>plain</b> , <b>precise</b> , <b>prominent</b> , <b>self-evident</b> , show, show up, <b>significantly</b> , <b>soberly</b> , <b>specific</b> , <b>straightforward</b> , <b>strong</b> , <b>sure</b> , <b>tangible</b> , <b>transparent</b> , <b>unambiguous</b> , <b>unambiguously</b> , <b>uncertain</b> , unclear, <b>undoubtedly</b> , <b>unequivocal</b> , <b>unequivocally</b> , <b>unspecific</b> , <b>vague</b> , <b>viewable</b> , <b>visibility</b> , visible, <b>visibly</b> , <b>visual</b> , <b>vividly</b> , well, <sup>1</sup> <b>woolly</b>
---

Table 1: 81 LUs invoking the Obviousness frame according to the new FrameNet+. New LUs (bold) have been added using the method of paraphrasing and human-vetting described in Section 4.

# Semantic Role Labeling

SRL is the task of automatically finding the semantic roles of each argument of each predicate in a sentence.

Most state-of-the-art approaches to SRL use supervised machine learning, with FrameNet and PropBank providing training and test sets and defining what counts as a predicate and what the roles are.

# Primitive Decomposition of Predicates

One way of thinking about semantic is that they help us define the roles that arguments play in a **decompositional** way, based on finite lists of thematic roles.

1. Jim killed his philodendron.  
Jim did something to cause his philodendron to become not alive.

$KILL(x,y) \Leftrightarrow CAUSE(x, BECOME(NOT(ALIVE(y))))$

2. John opened the door.  $\Rightarrow CAUSE(John, BECOME(OPEN(door)))$
3. The door opened.  $\Rightarrow BECOME(OPEN(door))$

The door is open.  $\Rightarrow OPEN(door)$

# Conceptual dependency primitives

Primitive	Definition
ATRANS	The abstract transfer of possession or control from one entity to another
PTRANS	The physical transfer of an object from one location to another
MTRANS	The transfer of mental concepts between entities or within an entity
MBUILD	The creation of new information within an entity
PROPEL	The application of physical force to move an object
MOVE	The integral movement of a body part by an animal
INGEST	The taking in of a substance by an animal
EXPEL	The expulsion of something from an animal
SPEAK	The action of producing a sound
ATTEND	The action of focusing a sense organ

# Neo-Davidsonian Event with Primitives

The waiter brought Mary the check.

$$\exists x,y \text{ Atrans}(x) \wedge \text{Actor}(x, \text{Waiter}) \wedge \text{Object}(x, \text{Check}) \wedge \text{To}(x, \text{Mary})$$
$$\wedge \text{Ptrans}(y) \wedge \text{Actor}(y, \text{Waiter}) \wedge \text{Object}(y, \text{Check}) \wedge \text{To}(y, \text{Mary})$$



# The Decompositional Semantics Initiative

*Rapid, simple, commonsensical annotations of meaning*

## *About*

The **Decompositional Semantics Initiative (Decomp)** collects and models simple, commonsensical annotations of meaning inspired by linguistic theory.

Traditional semantic annotation frameworks generally define complex, often exclusive category systems that require highly trained annotators to build. And in spite of their high quality for the cases they are designed to handle, these frameworks can be brittle to deviations from prototypical instances of a category.

<http://decomp.io>



# Conclusions

Semantic roles are abstract models of the role an argument plays in the event described by the predicate.

Thematic roles are a model of semantic roles based on a single finite list of roles.

Per-verb semantic role lists and proto-agent/proto-patient, are implemented in PropBank and FrameNet.

Semantic role labeling is the task of assigning semantic role labels to the constituents of a sentence.

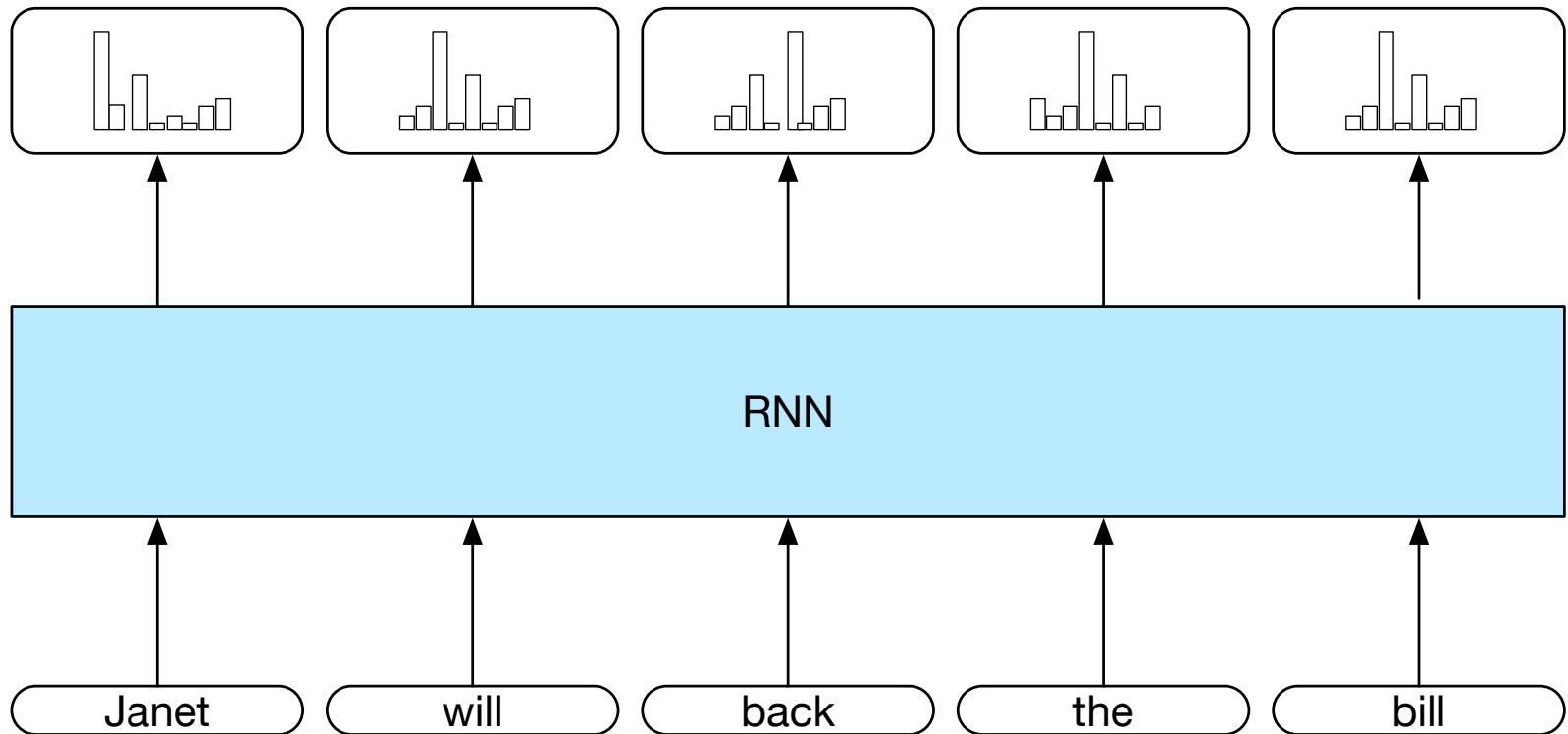
Cool new research directions in finding event primitives.

# Encoder-Decoder Models

JURAFSKY AND MARTIN CHAPTER 10

# Review: Recurrent Neural Networks (RNNs)

RNNs can be used for language modeling and sequence labeling. **Transduction** is the general process of taking in an input sequence and transforming it into output sequences in a one-to-one fashion.

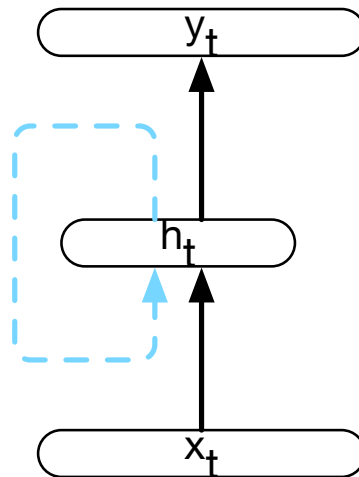


# Review: Recurrent Neural Networks (RNNs)

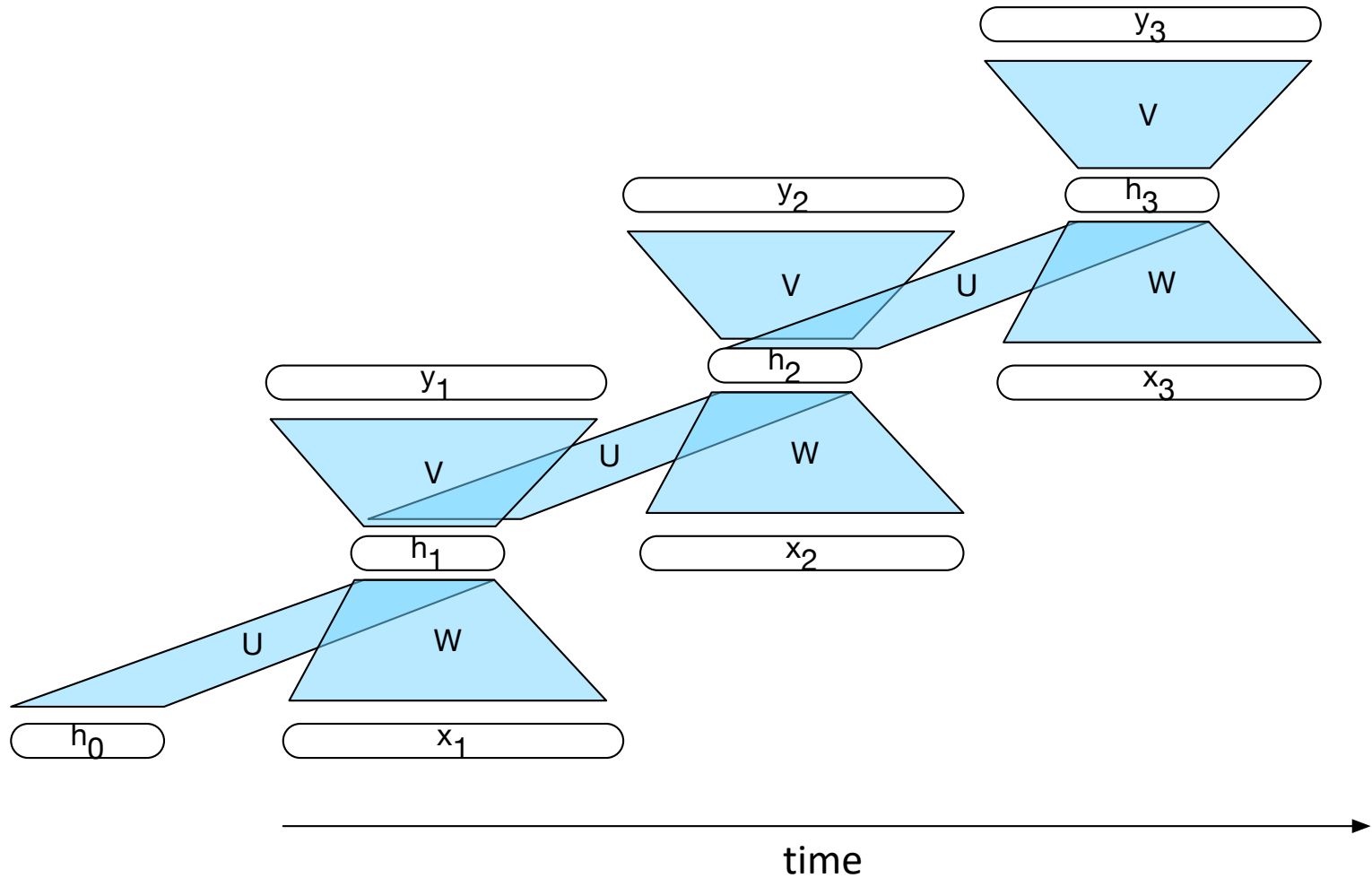
A recurrent neural network (RNN) is any network that contains a cycle within its network.

In such networks the value of a unit can be dependent on earlier outputs as an input.

RNNs have proven extremely effective when applied to NLP.



# Review: Unrolled RNN



# Review: Recurrent Neural Language Models

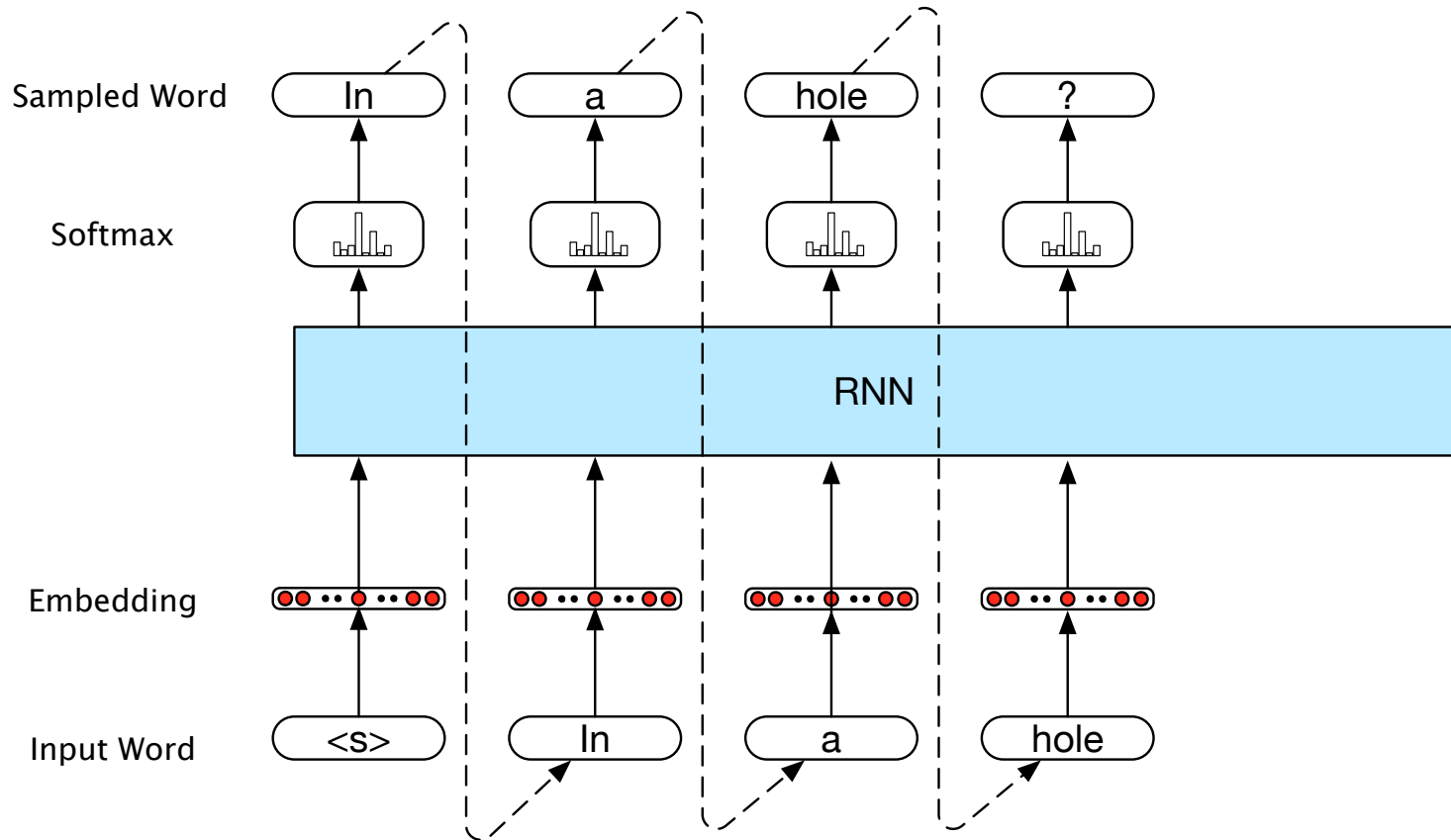
Unlike n-gram LMs and feedforward networks with sliding windows, RNN LMs don't use a fixed size context window.

They predict the next word in a sequence by using the current word and the previous hidden state as input.

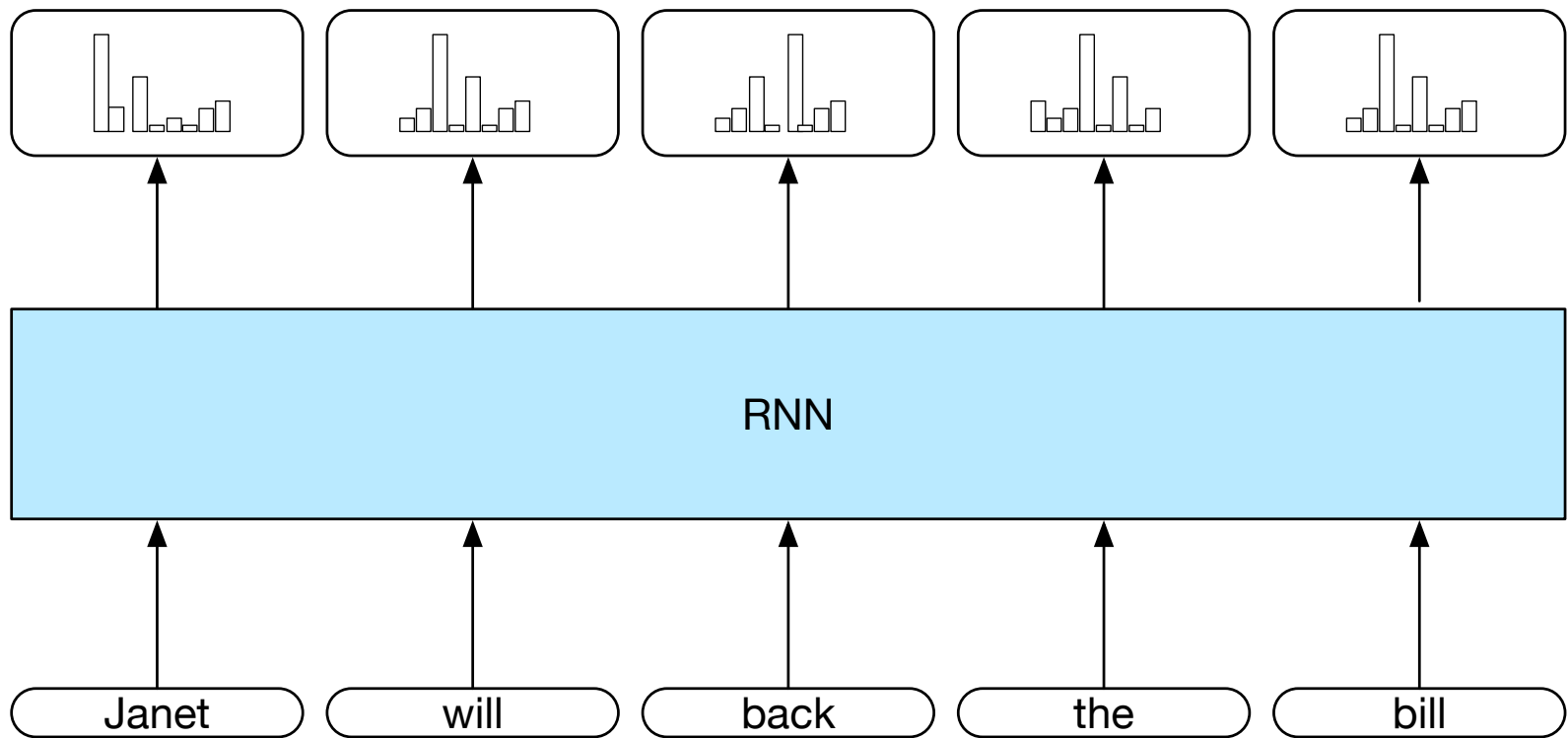
The hidden state embodies information about all of the preceding words all the way back to the beginning of the sequence.

Thus they can potentially take more context into account than n-gram LMs and NN LMs that use a sliding window.

# Generation with an RNN LM

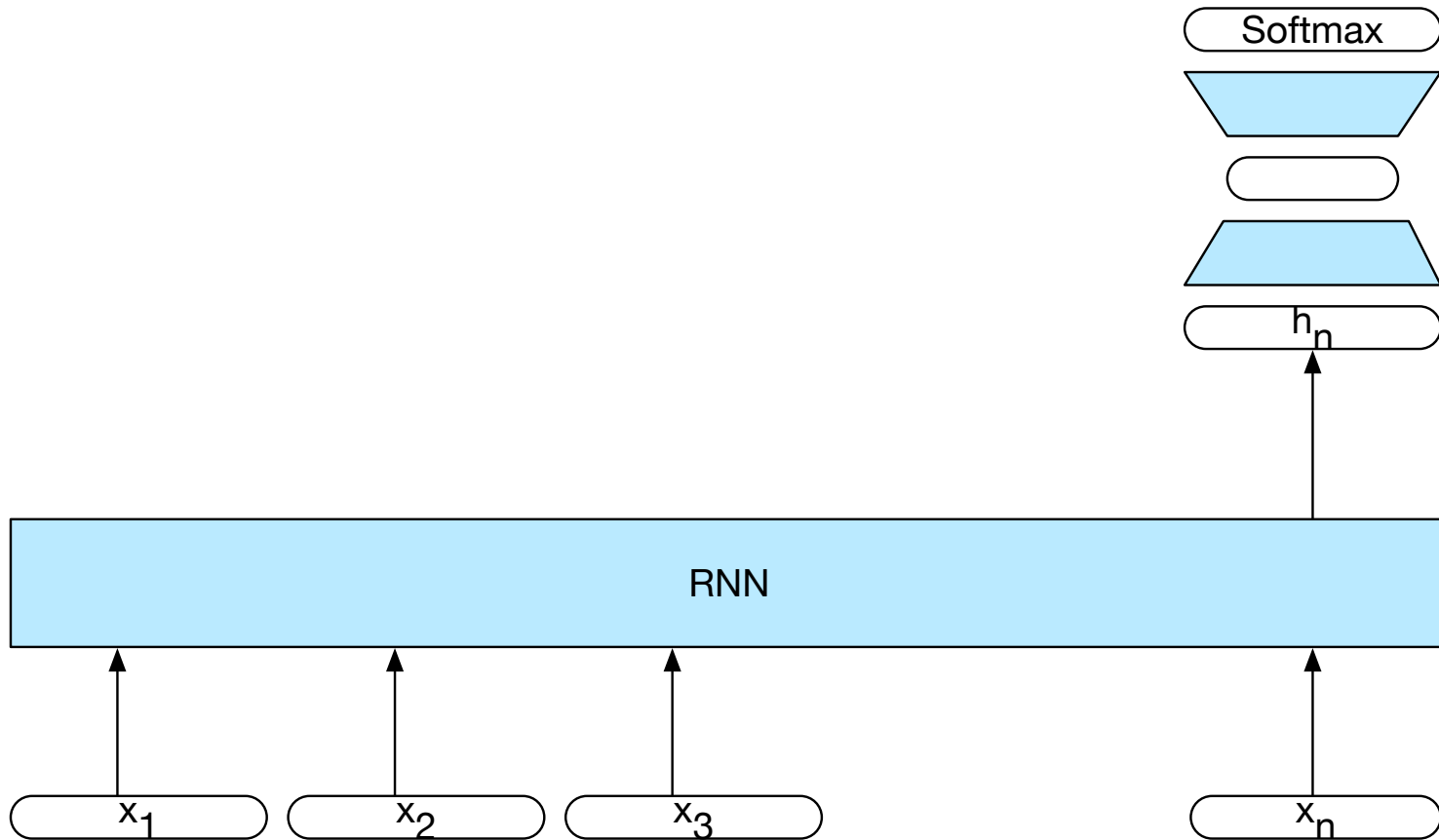


# Tag Sequences





# Sequence Classifiers



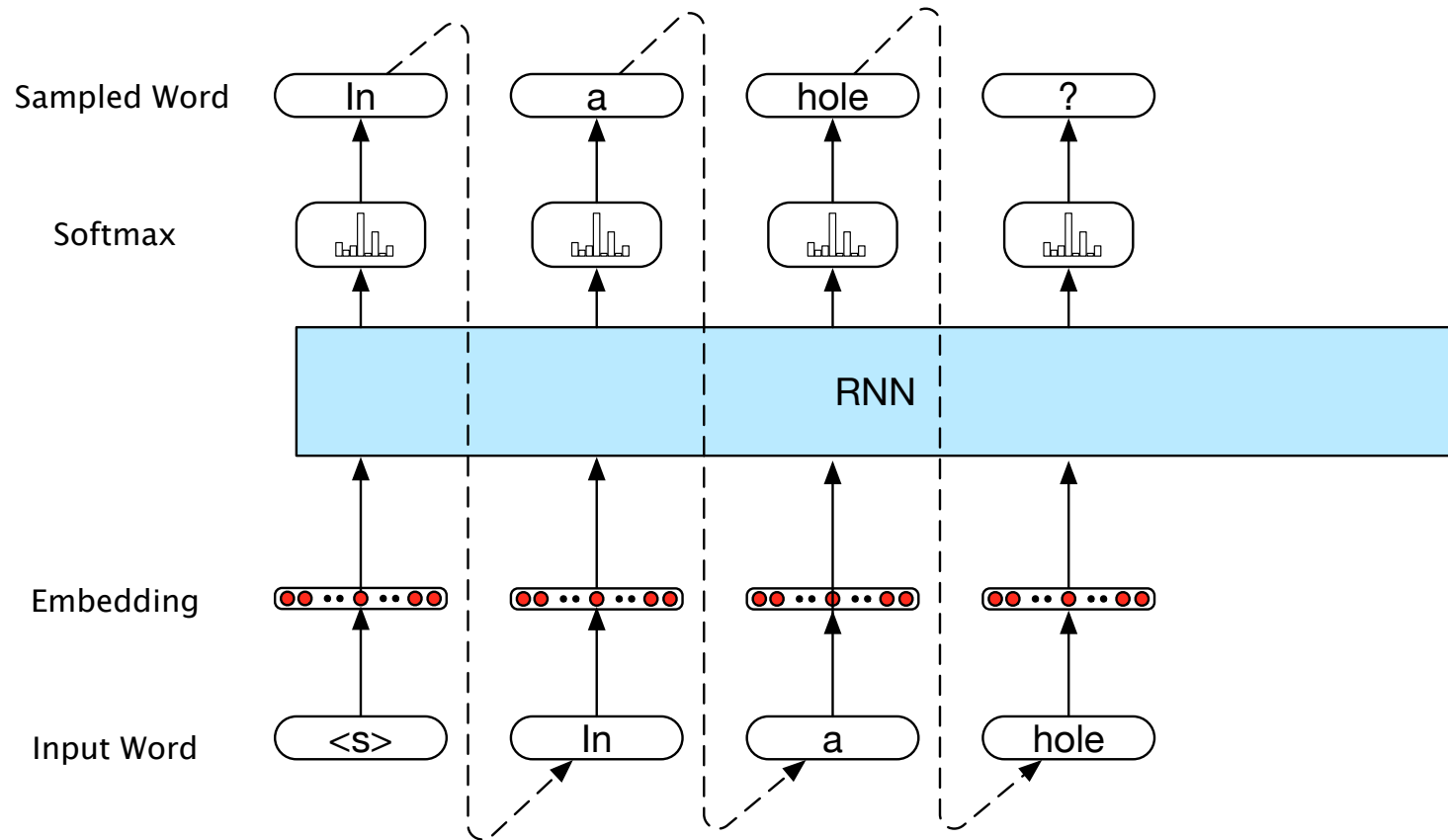
# Encoder-Decoder networks

**Encoder-decoder networks** are a kind of **sequence-to-sequence model**. Unlike vanilla RNNs, they can generate contextually appropriate, **arbitrary length**, output sequences.

They are useful for a wide range of NLP applications including:

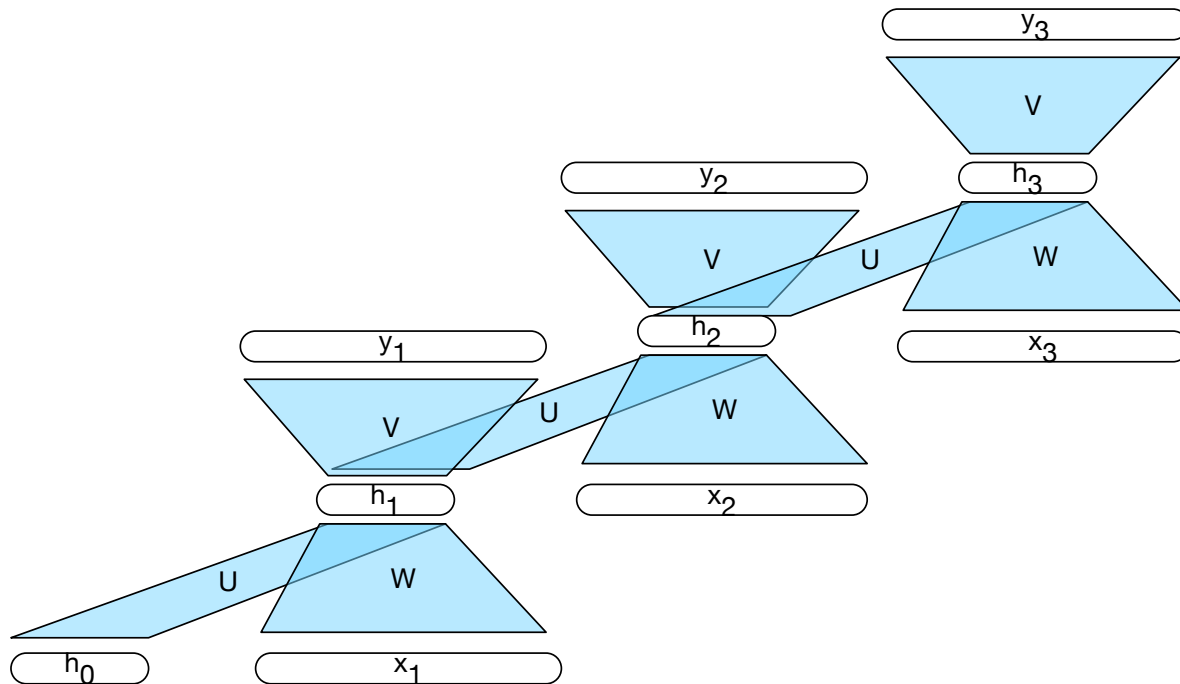
1. Machine Translation
2. Automatic summarization
3. Question answering
4. Dialog modelling

# Auto-Regressive Generation with an RNN LM

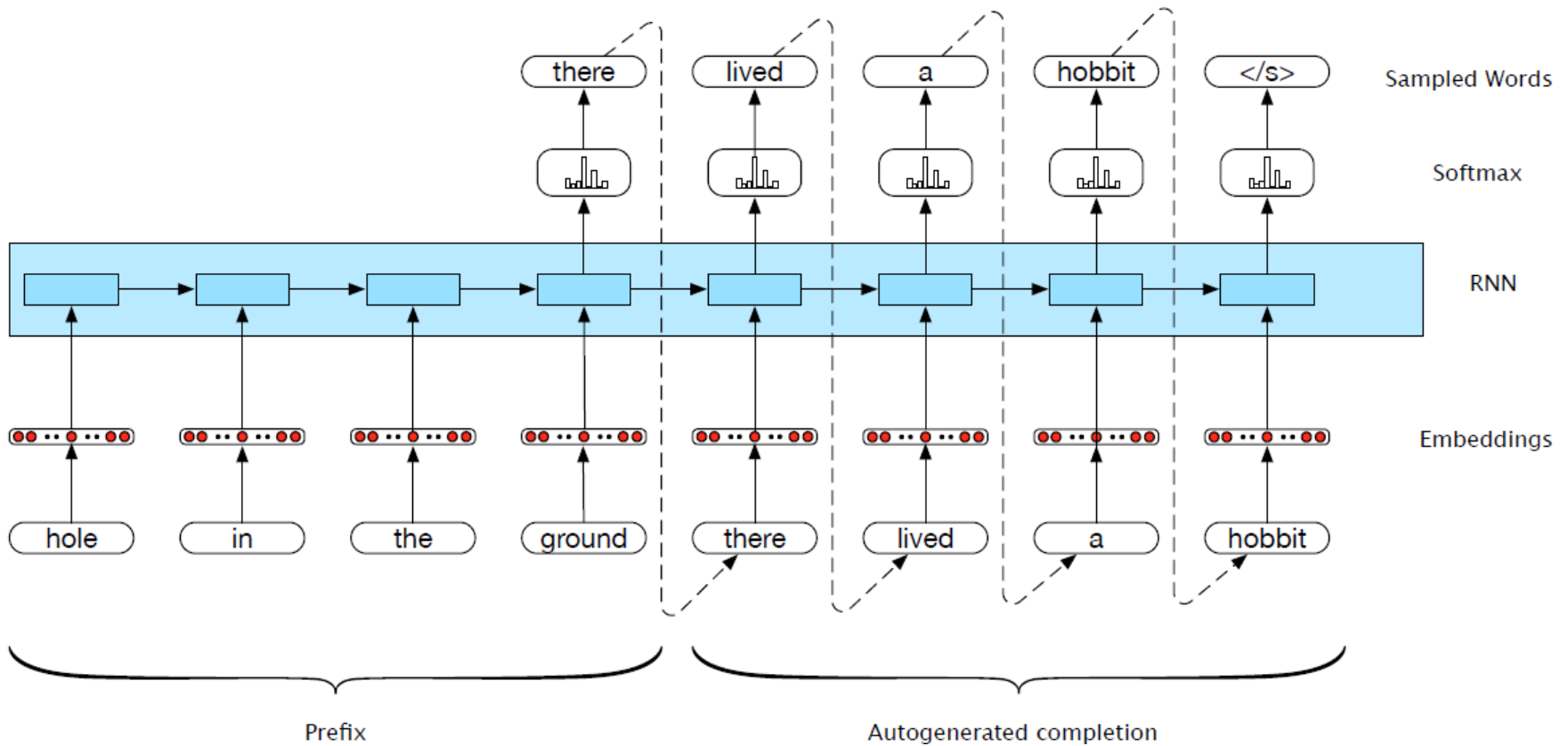


# Recall: autoregressive generation

- $h_t = g(Uh_{t-1} + Wx_t)$ ,  $y_t = f(Vh_t)$
- $f$  is a softmax over the set of possible outputs



# Generation with prefix



# Machine Translation (MT)

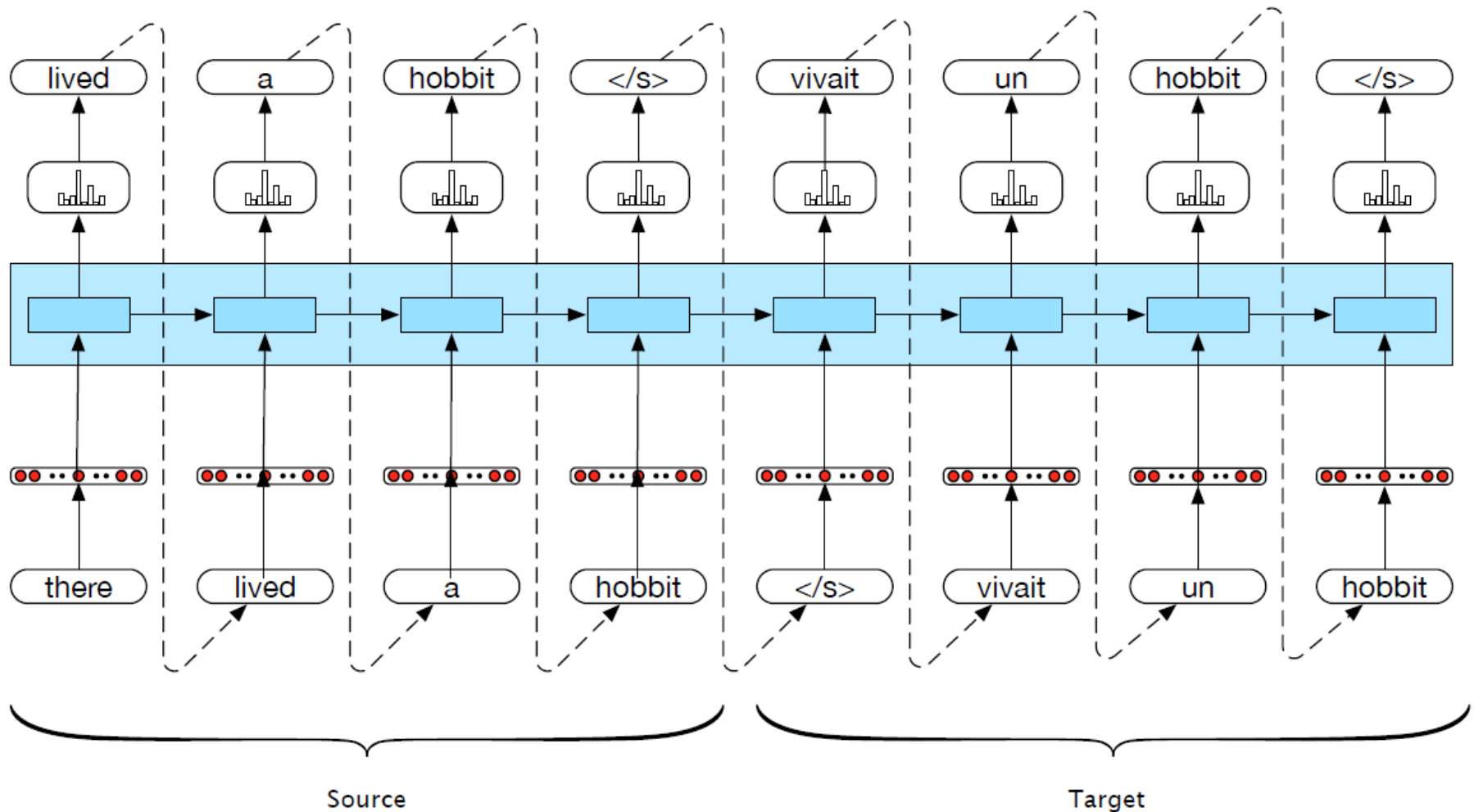
MT is the task of automatically translating sentences from one language into another.

We use bilingual parallel texts to train MT systems – pairs of **source-target** sentences that are translations of each other.

To extend LMs and autoregressive generation to MT, we will:

1. Add an end-of-sentence marker to each source sentence. Concatenate the target sentence to it.
2. Train an RNN LM based on this combined data.
3. To translate, simply treat the input sentence as a prefix, create a hidden state representation for it (**encoding step**).
4. Use the hidden state produced by the encoder to then start generating (**decoding step**)

# Machine translation



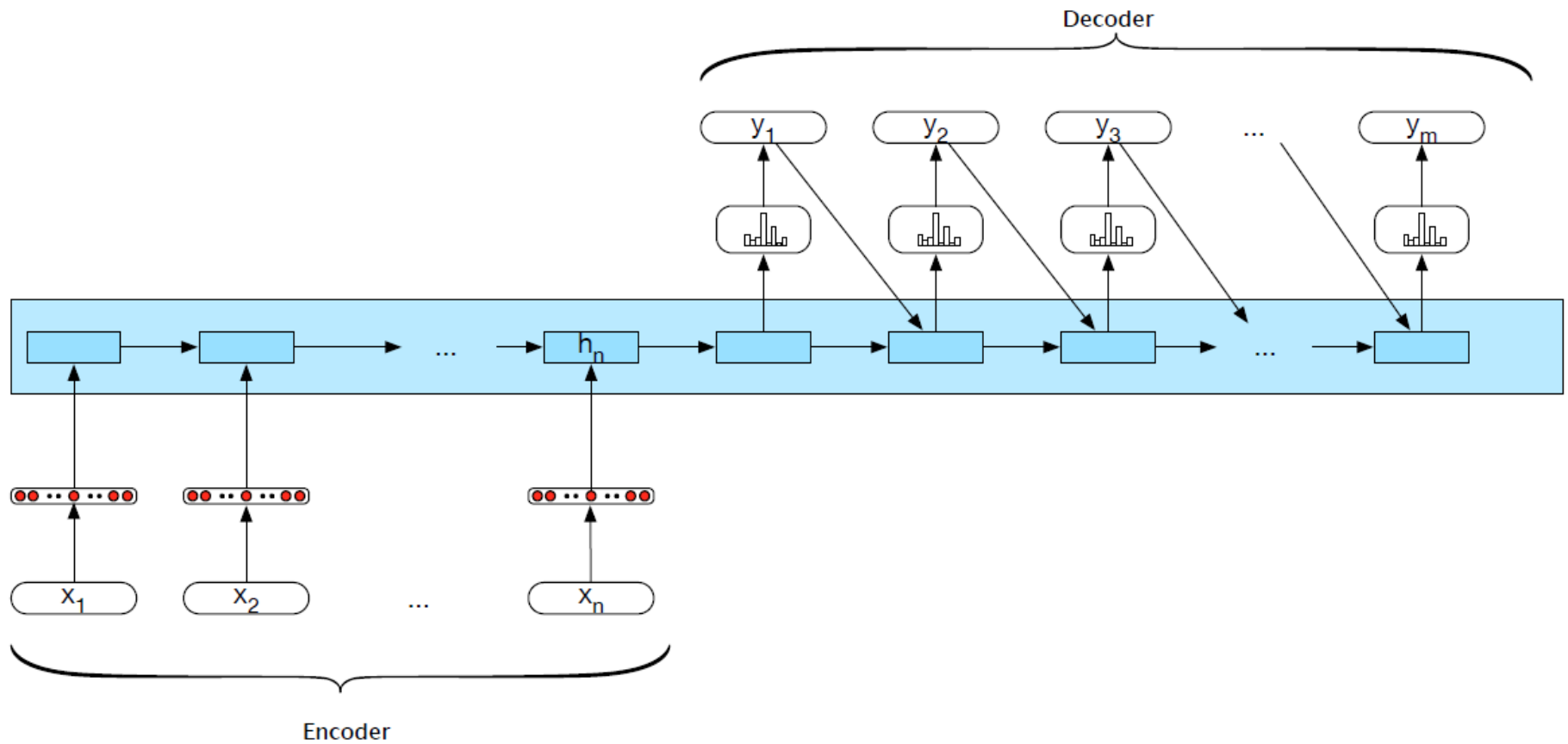
# Encoder-Decoder Networks

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  $y^m_1$ .

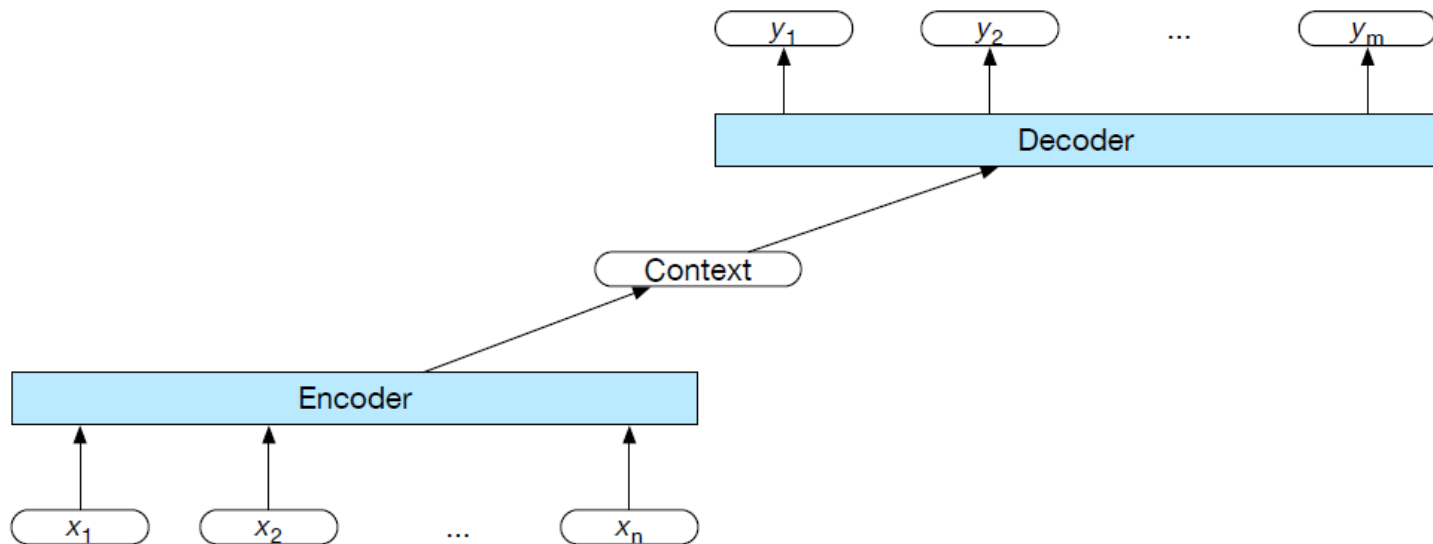


# Encoder-decoder networks



# 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

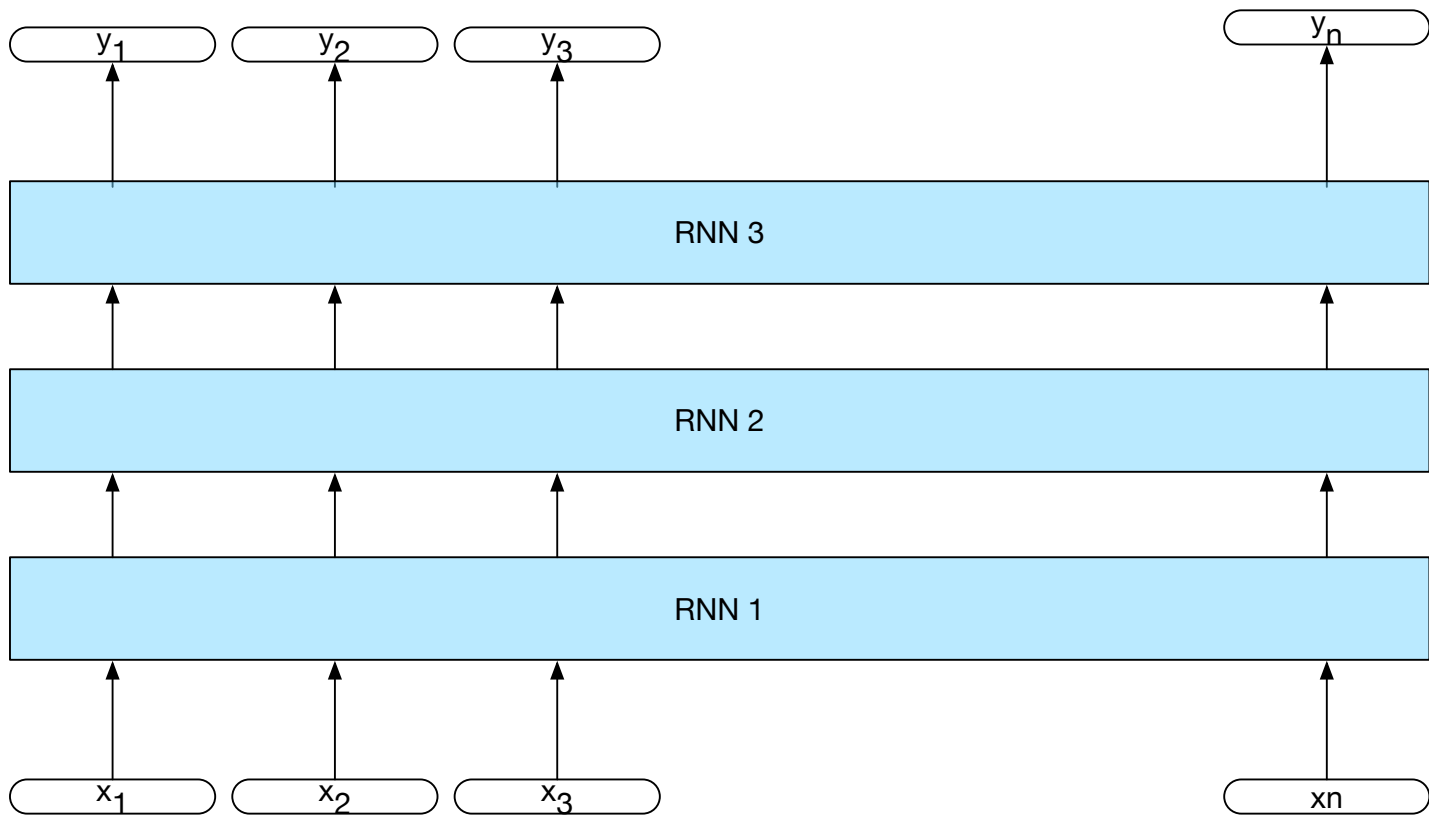


# Encoder

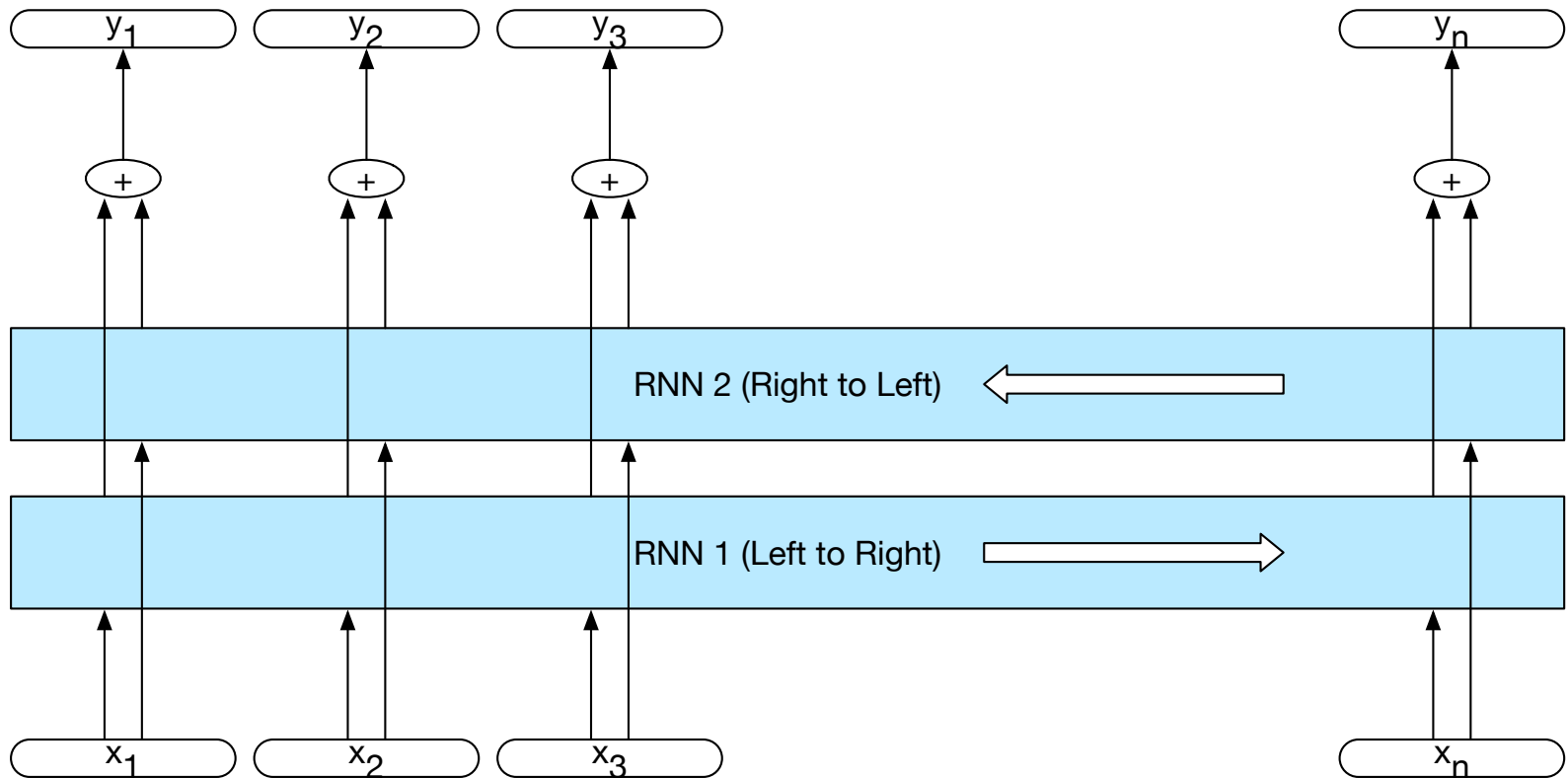
Pretty much any kind of RNN or its variants can be used as an encoder. Researchers have used simple RNNs, LSTMs, GRUs, or even convolutional networks.

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.

$$\begin{array}{l} \text{Encoder} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \text{Decoder} \end{array} \quad \begin{array}{l} c = h_n^e \\ h_0^d = c \\ \\ \\ h_t^d = g(\hat{y}_{t-1}, h_{t-1}^d) \\ z_t = f(h_t^d) \\ y_t = \text{softmax}(z_t) \end{array}$$

# 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 important 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 decoder

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

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

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

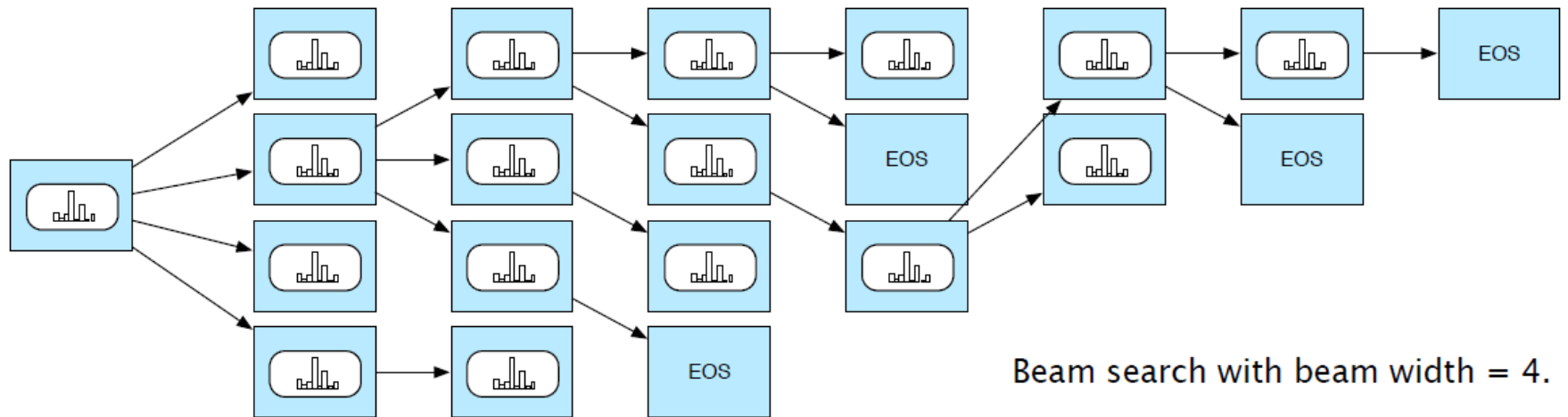
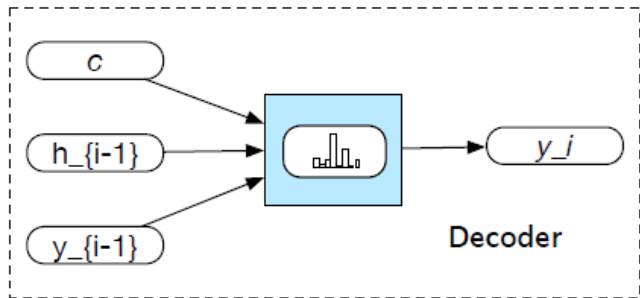


# Beam search

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.

0 1 2 3 4 5 6 7

# 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 mechanism

- 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

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

# Attention mechanism

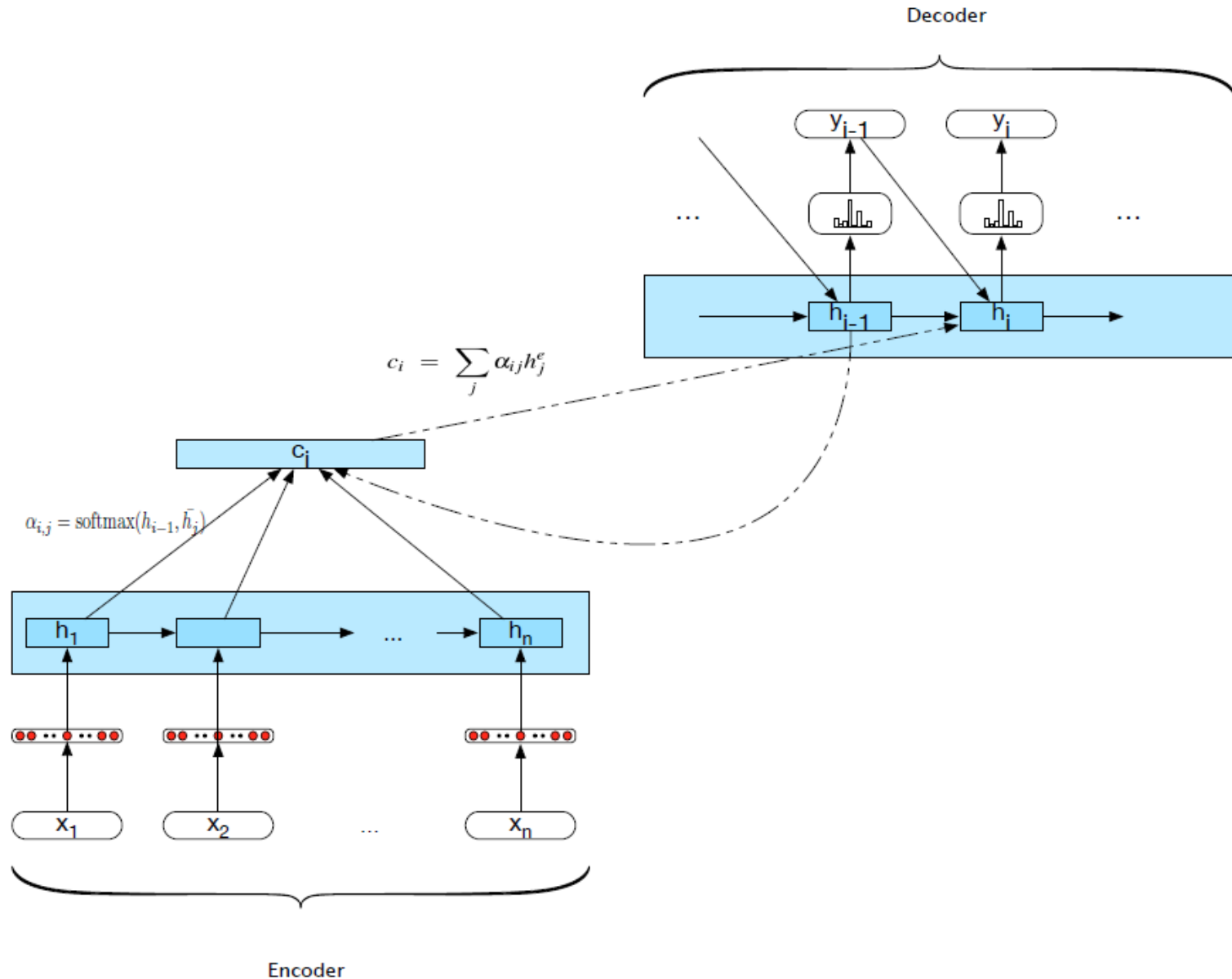
- To calculate  $c_i$ , first find relevance of each encoder hidden state to the decoder state. Call it  $score(h_{i-1}^d, h_j^e)$  for each encoder state  $j$ 
  - The  $score$  can simply be dot product, or be parameterized with weights
- Normalize them with a softmax to create a vector of weights  $\alpha_{i,j}$  that tells us the proportional relevance of each encoder hidden state  $j$  to the current decoder state  $i$

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

- 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?