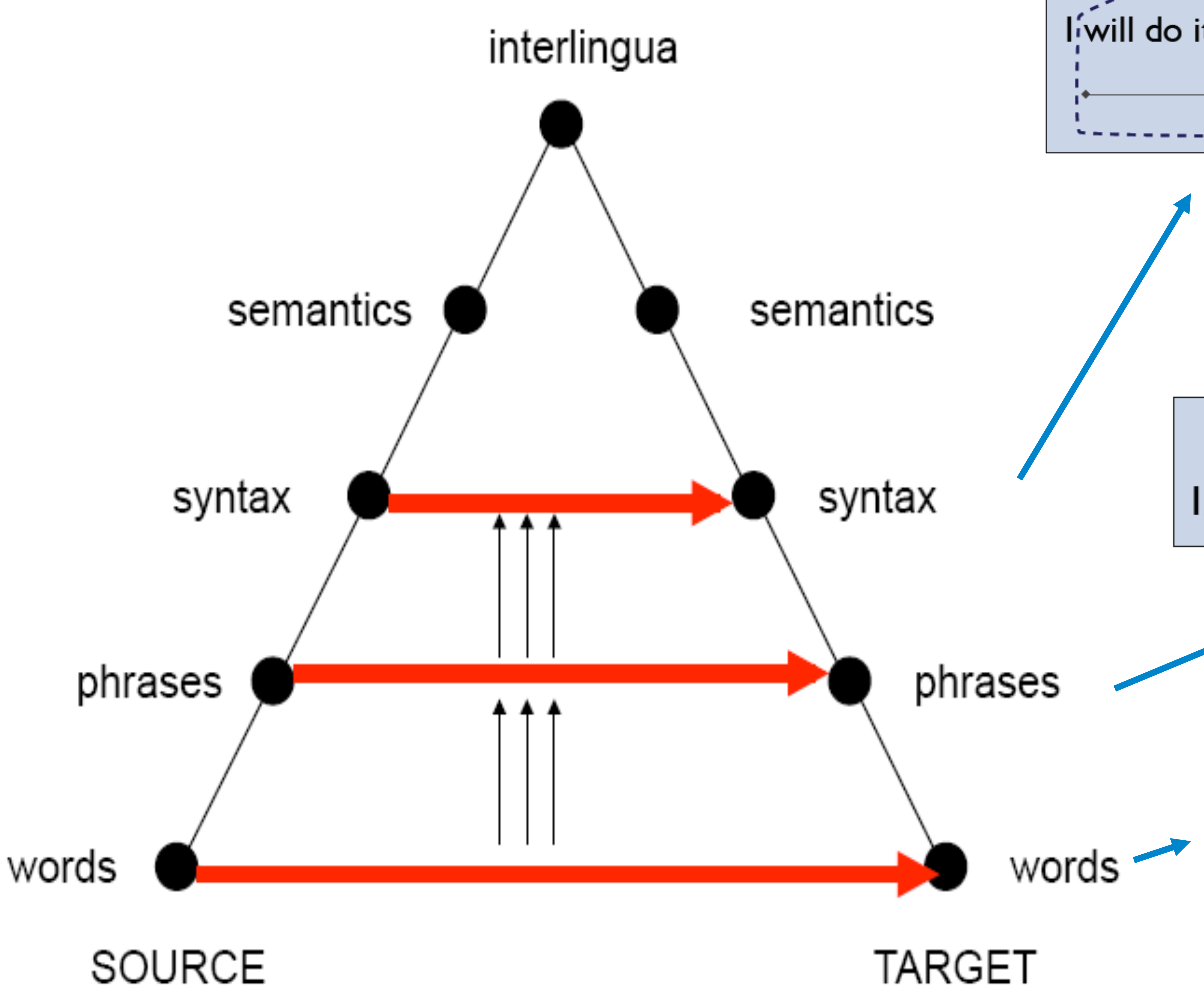
Lecture 13: Machine Translation II

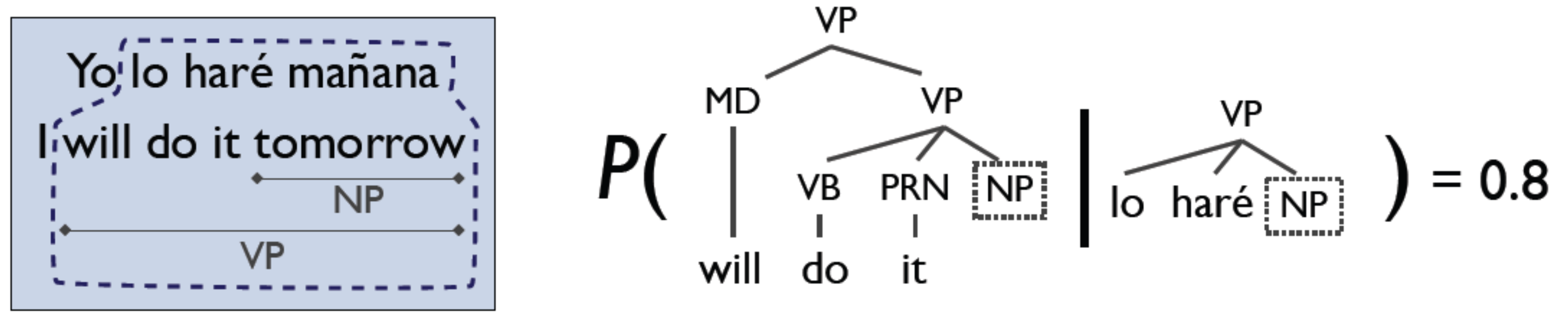
Alan Ritter

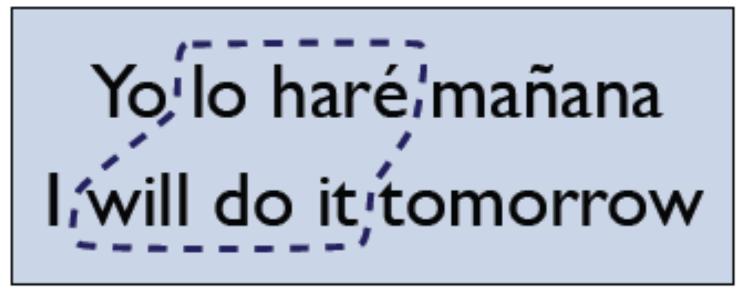
(many slides from Greg Durrett)

Syntactic MT

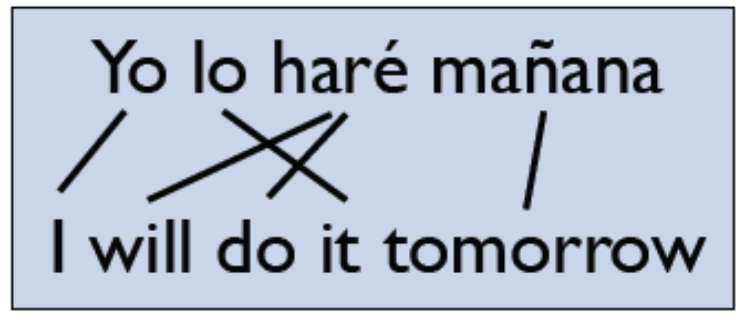
Levels of Transfer: Vauquois Triangle







English (E)	*P*(E | lo haré)
will do it	0.8
will do so	0.2



English (E)	*P*(E | mañana)
tomorrow	0.7
morning	0.3

- Is syntax a "better" abstraction than phrases?

Slide credit: Dan Klein

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs “parallel” trees in two languages simultaneously

$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow [\text{the},\ \text{la}]$

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow$ [the, la]

$DT \rightarrow$ [the, le]

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow$ [the, la]

$DT \rightarrow$ [the, le]

$NN \rightarrow$ [car, voiture]

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs “parallel” trees in two languages simultaneously

NP $\rightarrow$ [DT_1 JJ_2 NN_3; DT_1 NN_3 JJ_2]

DT $\rightarrow$ [the, la]

DT $\rightarrow$ [the, le]

NN $\rightarrow$ [car, voiture]

JJ $\rightarrow$ [yellow, jaune]

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow$ [the, la]

$DT \rightarrow$ [the, le]

$NN \rightarrow$ [car, voiture]

$JJ \rightarrow$ [yellow, jaune]

NP NP

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

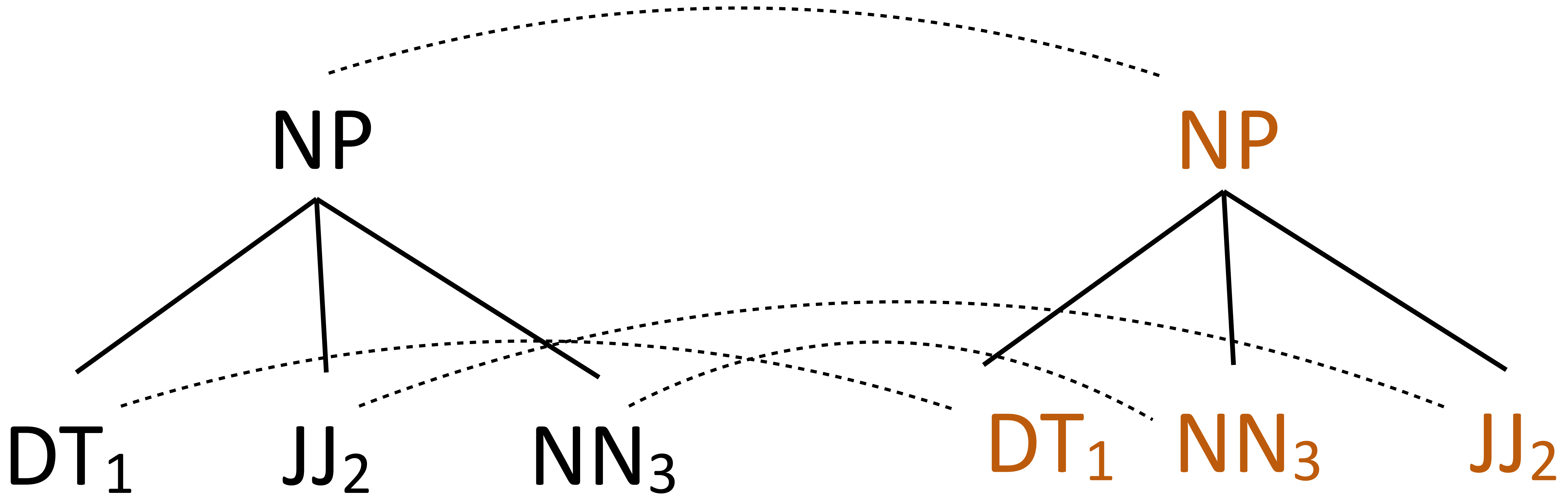
$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow$ [the, la]

$DT \rightarrow$ [the, le]

$NN \rightarrow$ [car, voiture]

$JJ \rightarrow$ [yellow, jaune]



Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

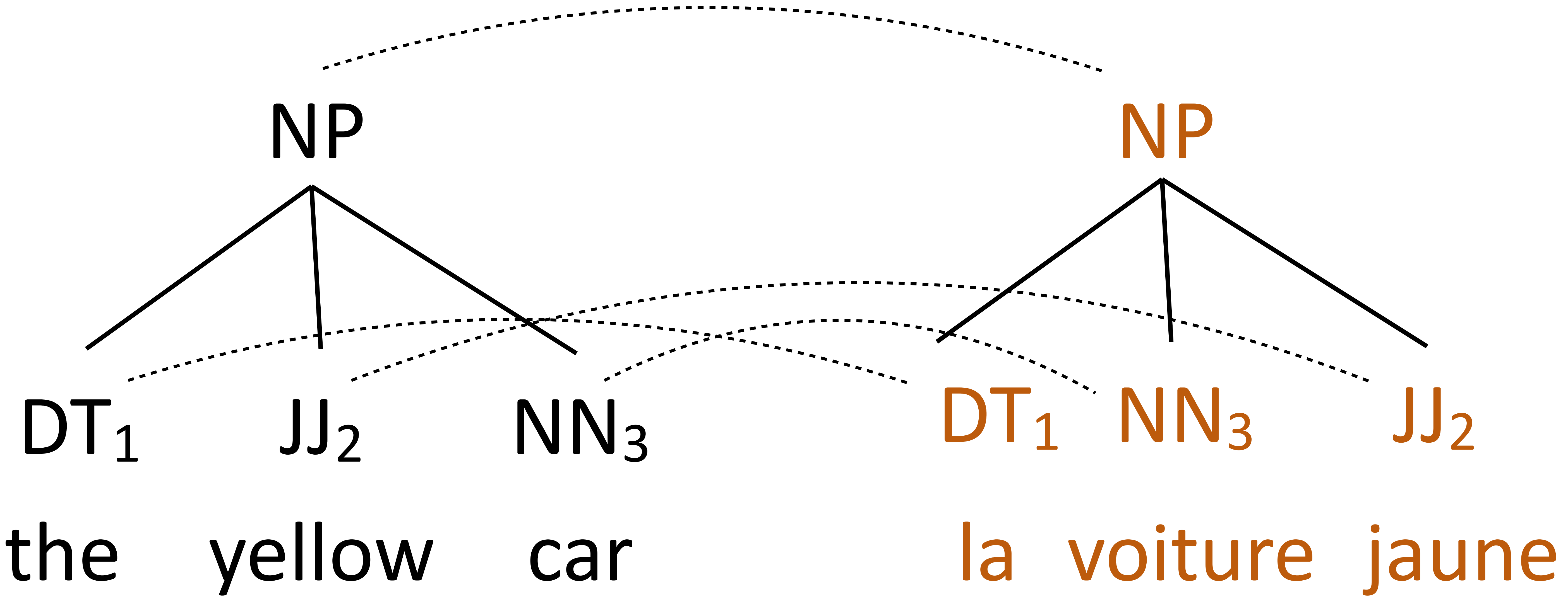
$NP \rightarrow [DT_1\ JJ_2\ NN_3;\ DT_1\ NN_3\ JJ_2]$

$DT \rightarrow$ [the, la]

$DT \rightarrow$ [the, le]

$NN \rightarrow$ [car, voiture]

$JJ \rightarrow$ [yellow, jaune]



Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

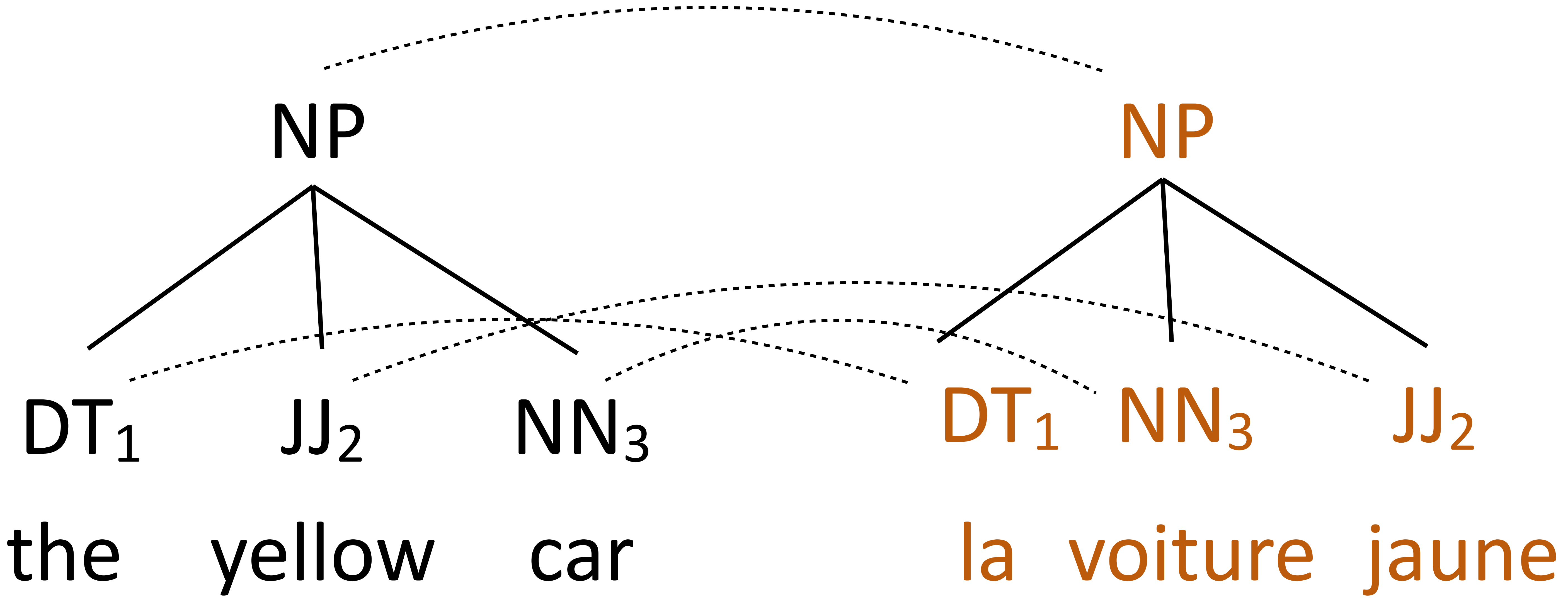
NP $\rightarrow$ [DT_1 JJ_2 NN_3; DT_1 NN_3 JJ_2]

DT $\rightarrow$ [the, la]

DT $\rightarrow$ [the, le]

NN $\rightarrow$ [car, voiture]

JJ $\rightarrow$ [yellow, jaune]



- Assumes parallel syntax up to reordering

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

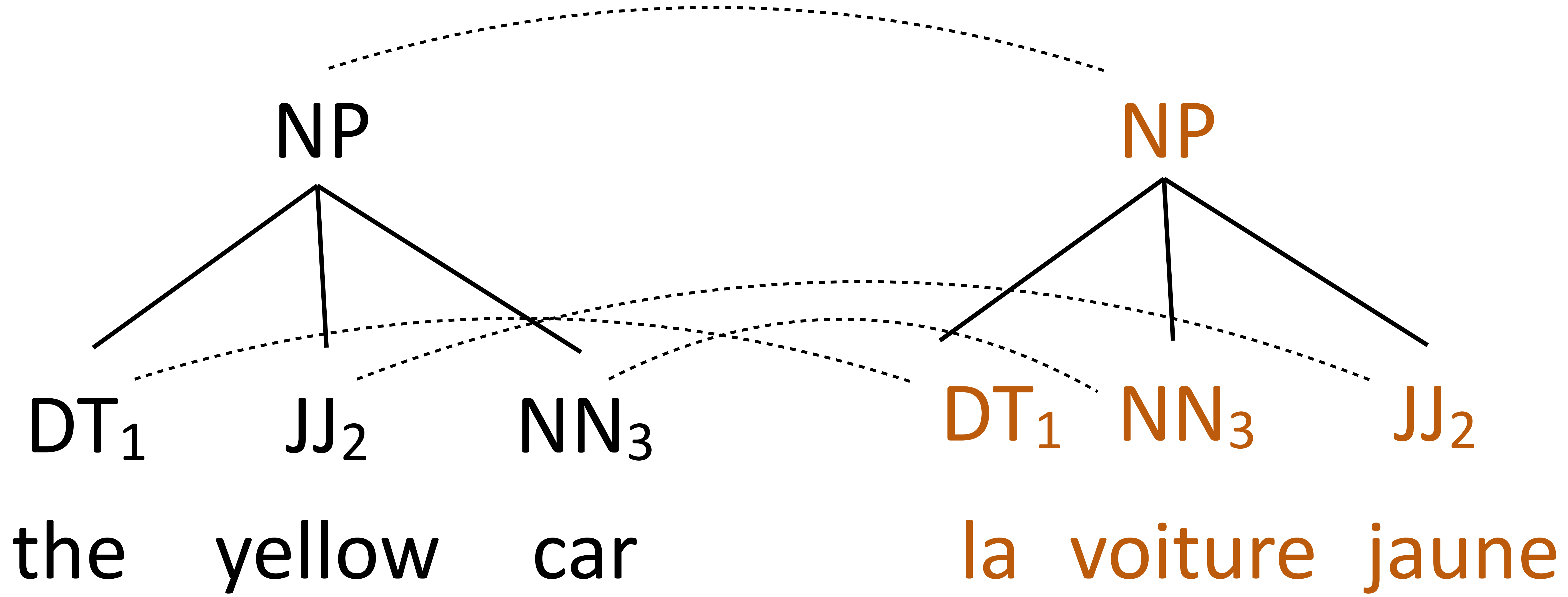
NP $\rightarrow$ [DT$_1$ JJ$_2$ NN$_3$; DT$_1$ NN$_3$ JJ$_2$]

DT $\rightarrow$ [the, la]

DT $\rightarrow$ [the, le]

NN $\rightarrow$ [car, voiture]

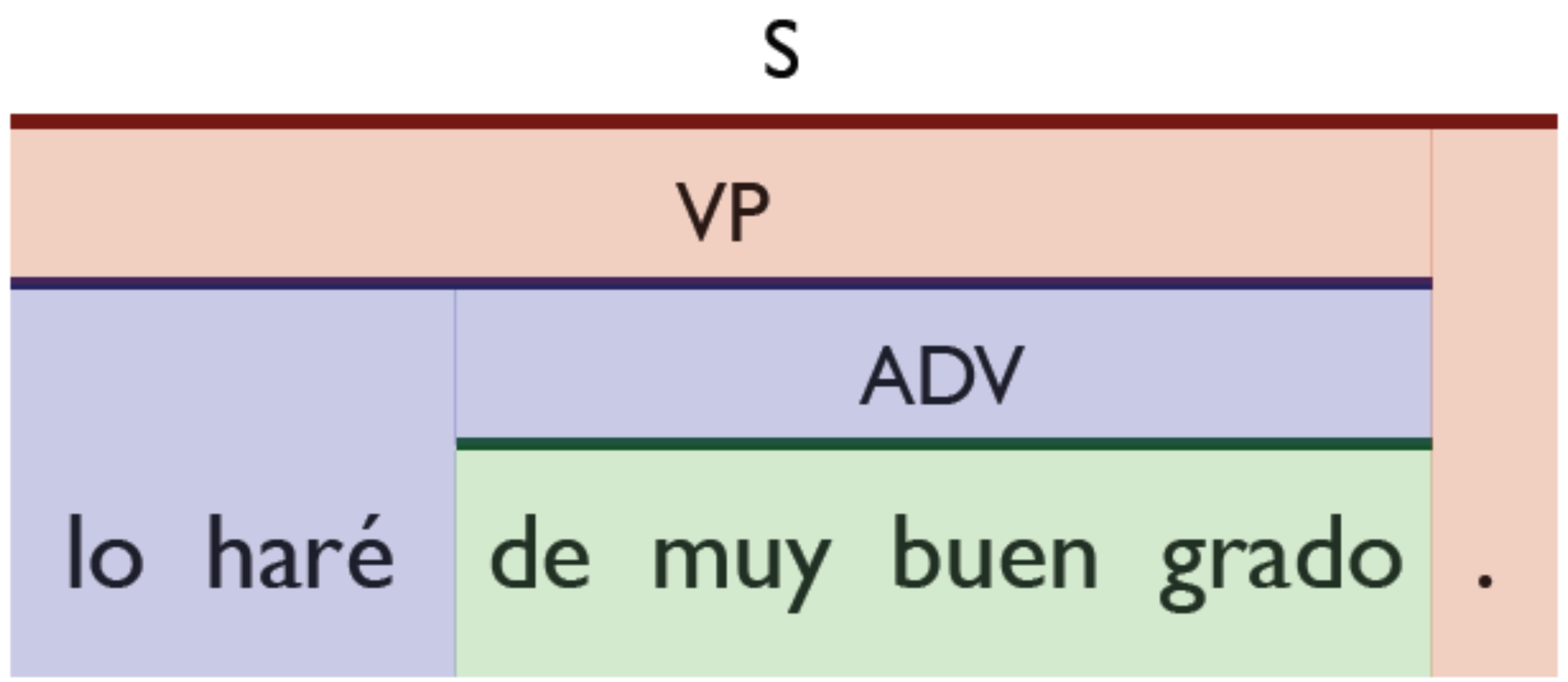
JJ $\rightarrow$ [yellow, jaune]



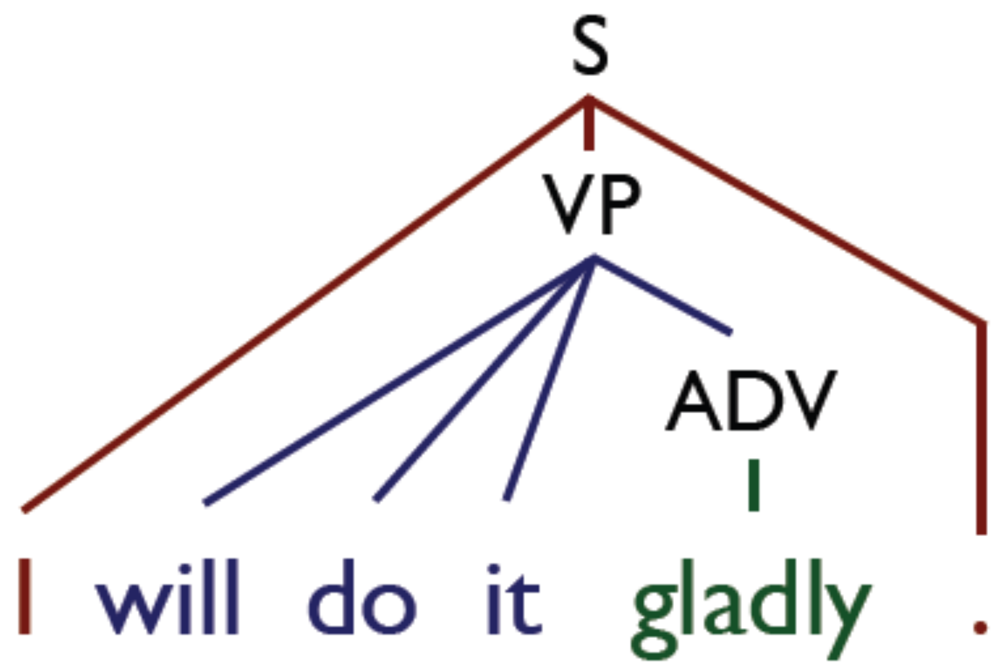
- Assumes parallel syntax up to reordering
- Translation = parse the input with "half" the grammar, read off other half

Syntactic MT

Input



Output



Grammar

S → ⟨ VP . ; I VP . ⟩ **OR** S → ⟨ VP . ; you VP . ⟩

VP → ⟨ lo haré ADV ; will do it ADV ⟩

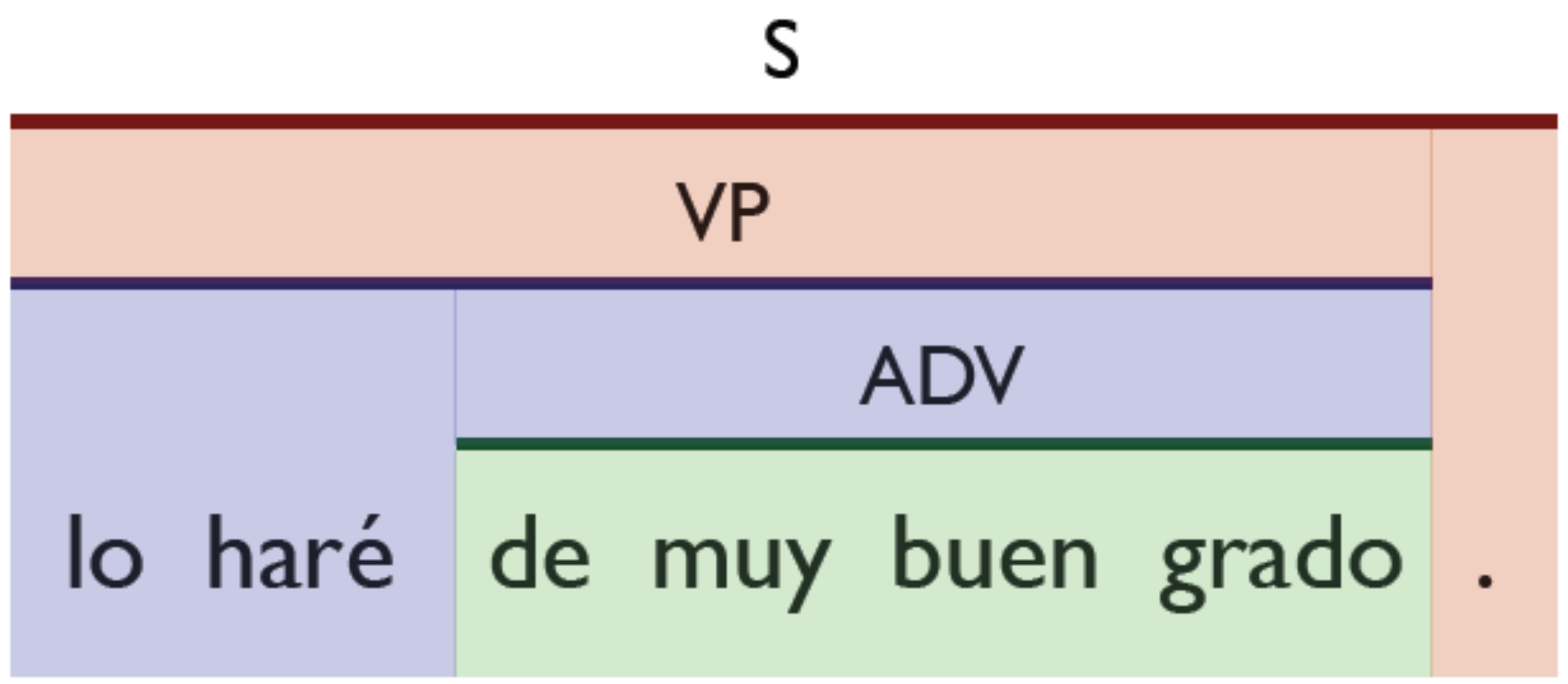
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩

ADV → ⟨ de muy buen grado ; gladly ⟩

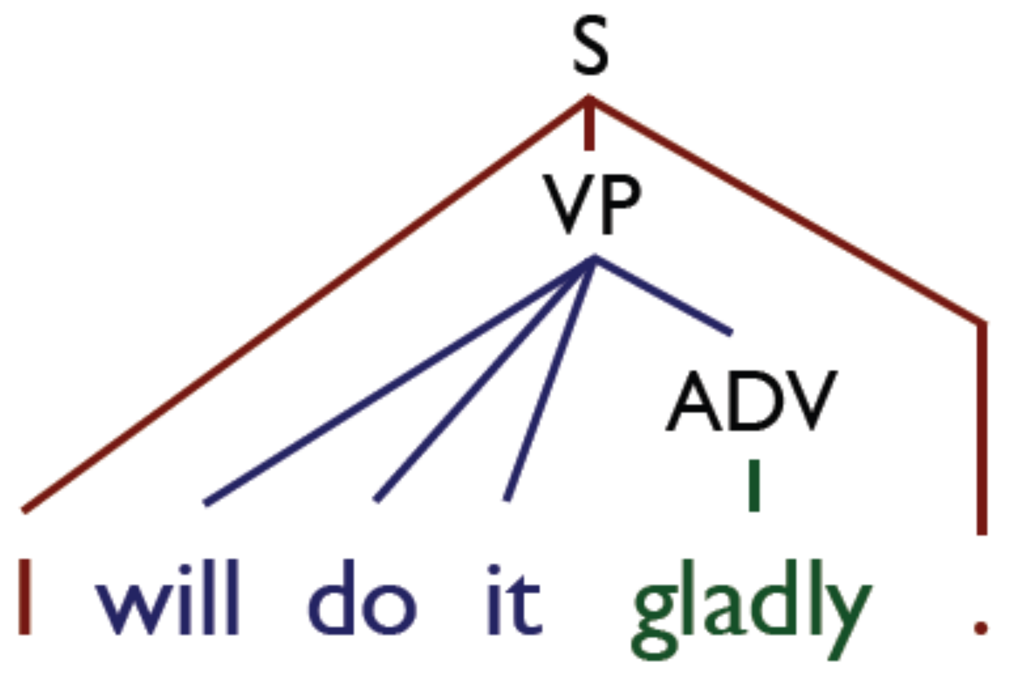
Slide credit: Dan Klein

Syntactic MT

Input



Output



Grammar

- Relax this by using lexicalized rules, like "syntactic phrases"

S → ⟨ VP . ; I VP . ⟩ **OR** S → ⟨ VP . ; you VP . ⟩

VP → ⟨ lo haré ADV ; will do it ADV ⟩

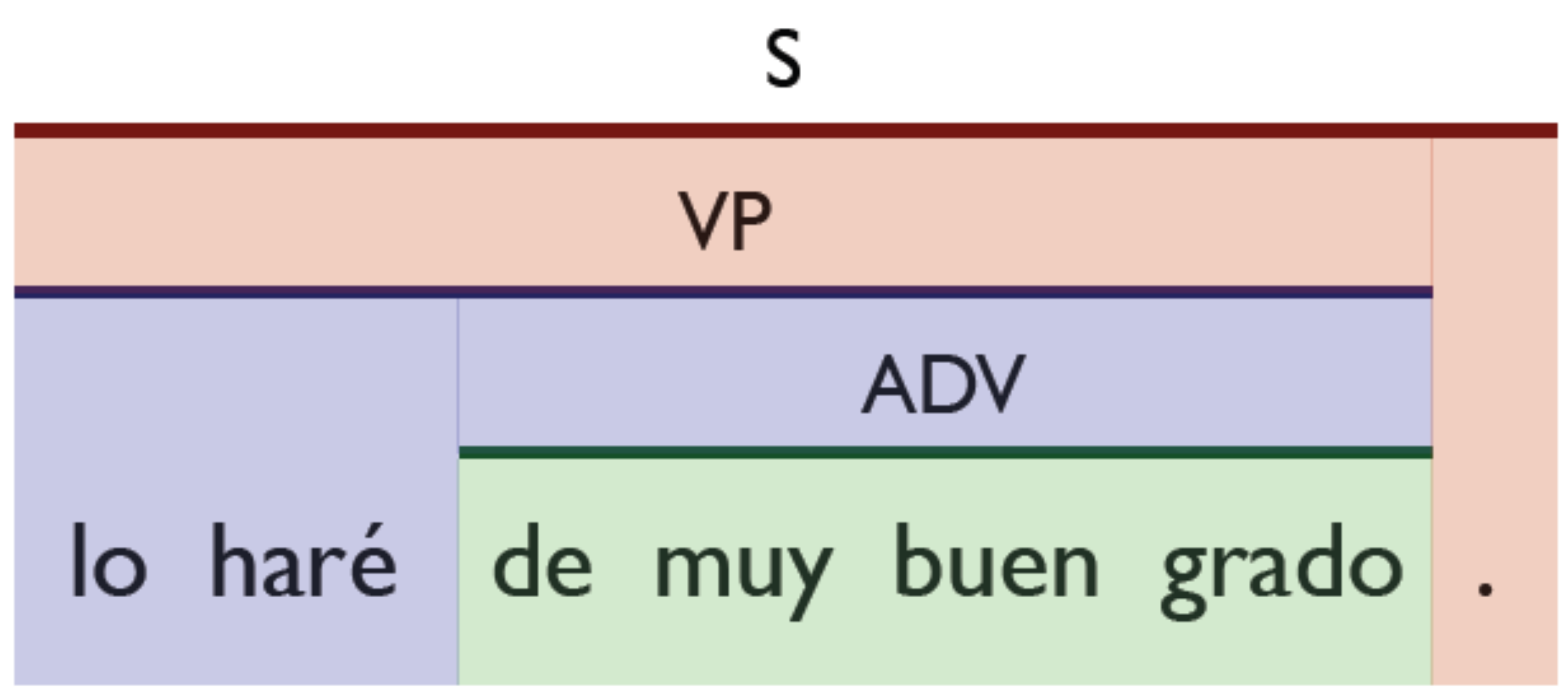
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩

ADV → ⟨ de muy buen grado ; gladly ⟩

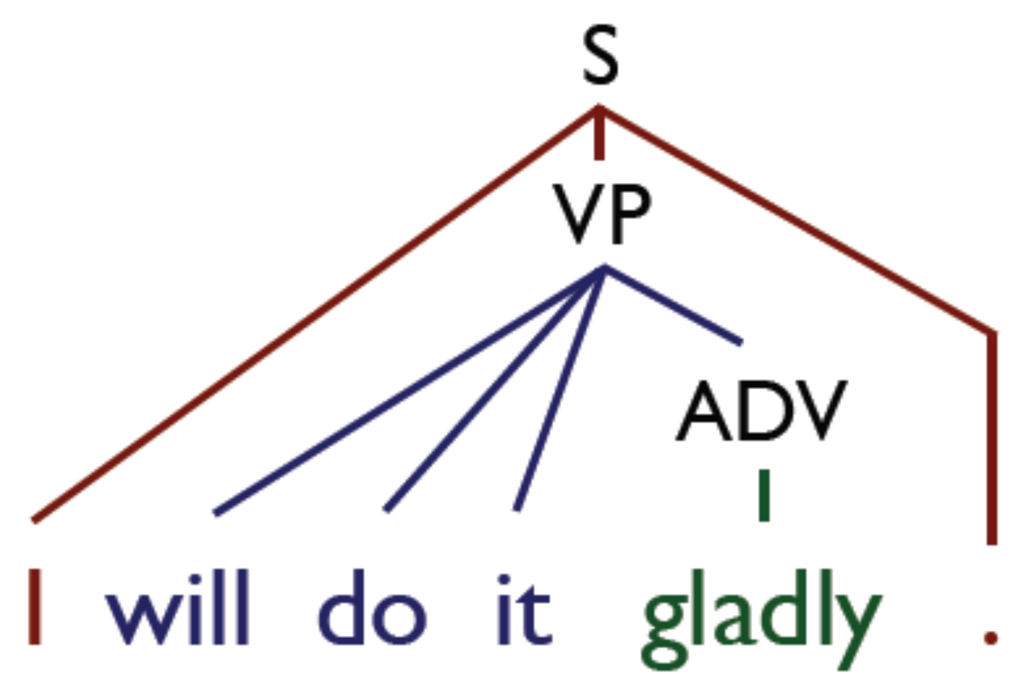
Slide credit: Dan Klein

Syntactic MT

Input



Output



Grammar

S → ⟨ VP . ; I VP . ⟩ **OR** S → ⟨ VP . ; you VP . ⟩

VP → ⟨ lo haré ADV ; will do it ADV ⟩

S → ⟨ lo haré ADV . ; I will do it ADV . ⟩

ADV → ⟨ de muy buen grado ; gladly ⟩

- Relax this by using lexicalized rules, like “syntactic phrases”
- Leads to HUGE grammars, parsing is slow

Slide credit: Dan Klein

Neural MT Details

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP

Sutskever et al. (2014)

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

Sutskever et al. (2014)

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Sutskever et al. (2014)

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

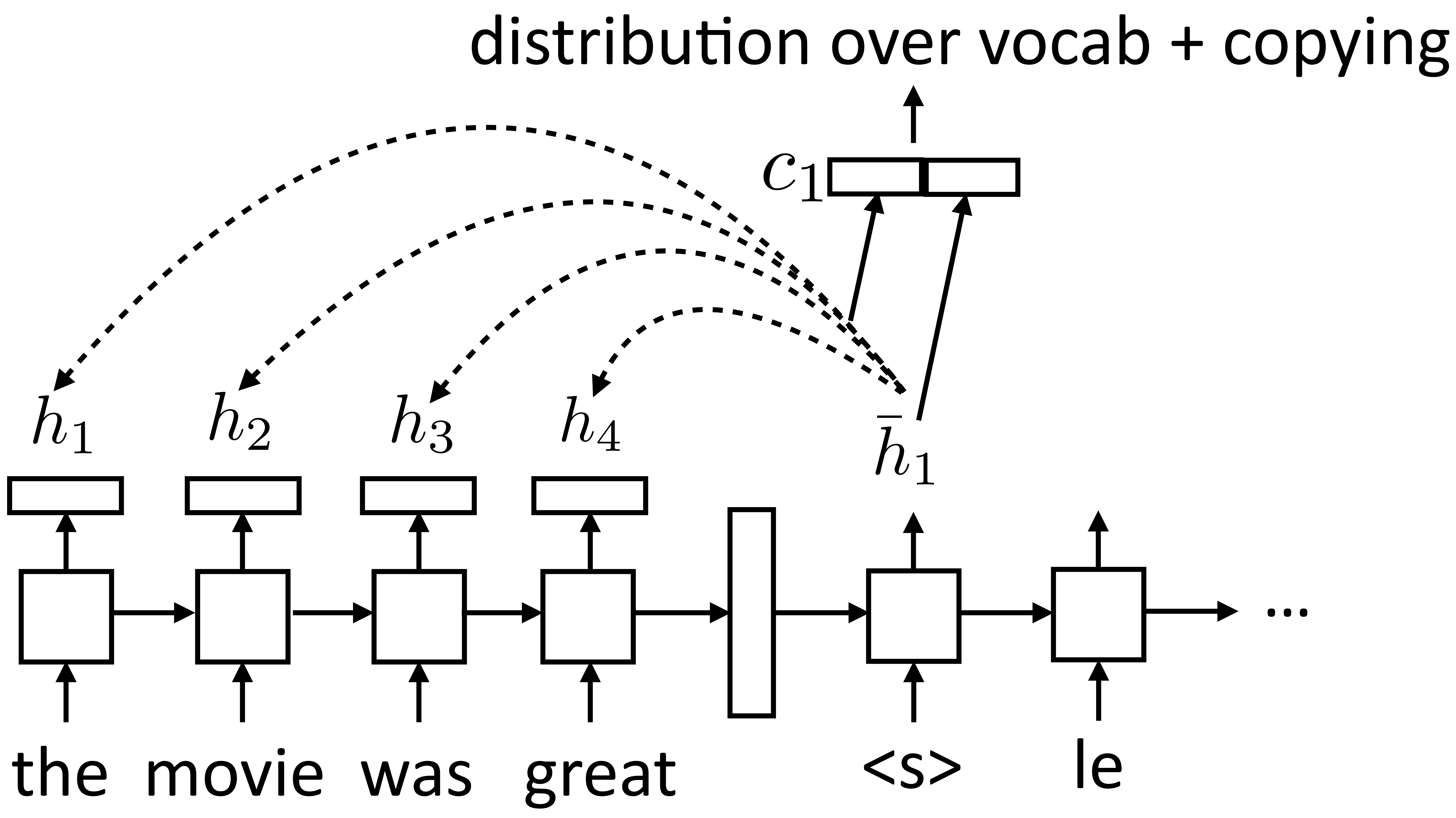
Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

- SOTA = 37.0 — not all that competitive…

Sutskever et al. (2014)

Encoder-Decoder MT

- Better model from seq2seq lectures: encoder-decoder with attention and copying for rare words



Results: WMT English-French

Results: WMT English-French

- 12M sentence pairs

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

Luong+ (2015) seq2seq ensemble with attention and rare word handling: **37.5** BLEU

- But English-French is a really easy language pair and there's *tons* of data for it! Does this approach work for anything harder?

Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but not really comparable across languages

Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but not really comparable across languages
- French, Spanish = easiest
  German, Czech = harder
  Japanese, Russian = hard (grammatically different, lots of morphology…)

MT Examples

src	In einem Interview sagte Bloom jedoch , dass er und Kerr sich noch immer lieben .
ref	However , in an interview , Bloom has said that he and *Kerr* still love each other .
best	In an interview , however , Bloom said that he and *Kerr* still love .
base	However , in an interview , Bloom said that he and **Tina** were still <unk> .

- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn't do this

Luong et al. (2015)

MT Examples

src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .
base	Because of the pressure **imposed by the European Central Bank and the Federal Central Bank with the strict austerity** imposed on the national economy in the face of the single currency , many people believe that the European project has gone too far .

- best = with attention, base = no attention

Luong et al. (2015)

MT Examples

Source	such changes in reaction conditions include , but are not limited to , an increase in temperature or change in ph .
Reference	所(such) 述(said) 反 应(reaction) 条 件(condition) 的(of) 改 变(change) 包 括(include) 但(but) 不(not) 限 于(limit) 温度(temperature) 的(of) 增加(increase) 或(or) pH 值(value) 的(of) 改变(change) 。
PBMT	中(in) 的(of) 这 种(such) 变 化(change) 的(of) 反 应(reaction) 条 件(condition) 包 括(include) , 但(but) 不(not) 限 于(limit) , 增加(increase) 的(of) 温度(temperature) 或(or) pH 变化(change) 。
NMT	这种(such) 反应(reaction) 条件(condition) 的(of) 变化(change) 包括(include) 但(but) 不(not) 限于(limit) pH 或(or) pH 的(of) 变化(change) 。

- NMT can repeat itself if it gets confused (pH or pH)
- Phrase-based MT often gets chunks right, may have more subtle ungrammaticalities

Zhang et al. (2017)

Rare Words: Word Piece Models

- Use Huffman encoding on a corpus, keep most common *k* (~10,000) character sequences for source and target

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is …

Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

Wu et al. (2016)

Rare Words: Word Piece Models

- Use Huffman encoding on a corpus, keep most common *k* (~10,000) character sequences for source and target

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is …

Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

- Captures common words and parts of rare words

Wu et al. (2016)

Rare Words: Word Piece Models

- Use Huffman encoding on a corpus, keep most common *k* (~10,000) character sequences for source and target

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is …

Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

- Captures common words and parts of rare words
- Subword structure may make it easier to translate

Wu et al. (2016)

Rare Words: Word Piece Models

- Use Huffman encoding on a corpus, keep most common *k* (~10,000) character sequences for source and target

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is ...

Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

- Captures common words and parts of rare words
- Subword structure may make it easier to translate
- Model balances translating and transliterating without explicit switching

Wu et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

Sennrich et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

Sennrich et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences

Sennrich et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences
- Merge the most frequent pair of adjacent characters

Sennrich et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences
- Merge the most frequent pair of adjacent characters

- Final size = initial vocab + num merges. Often do 10k - 30k merges

Sennrich et al. (2016)

Rare Words: Byte Pair Encoding

- Simpler procedure, based only on the dictionary
- Input: a dictionary of words represented as characters

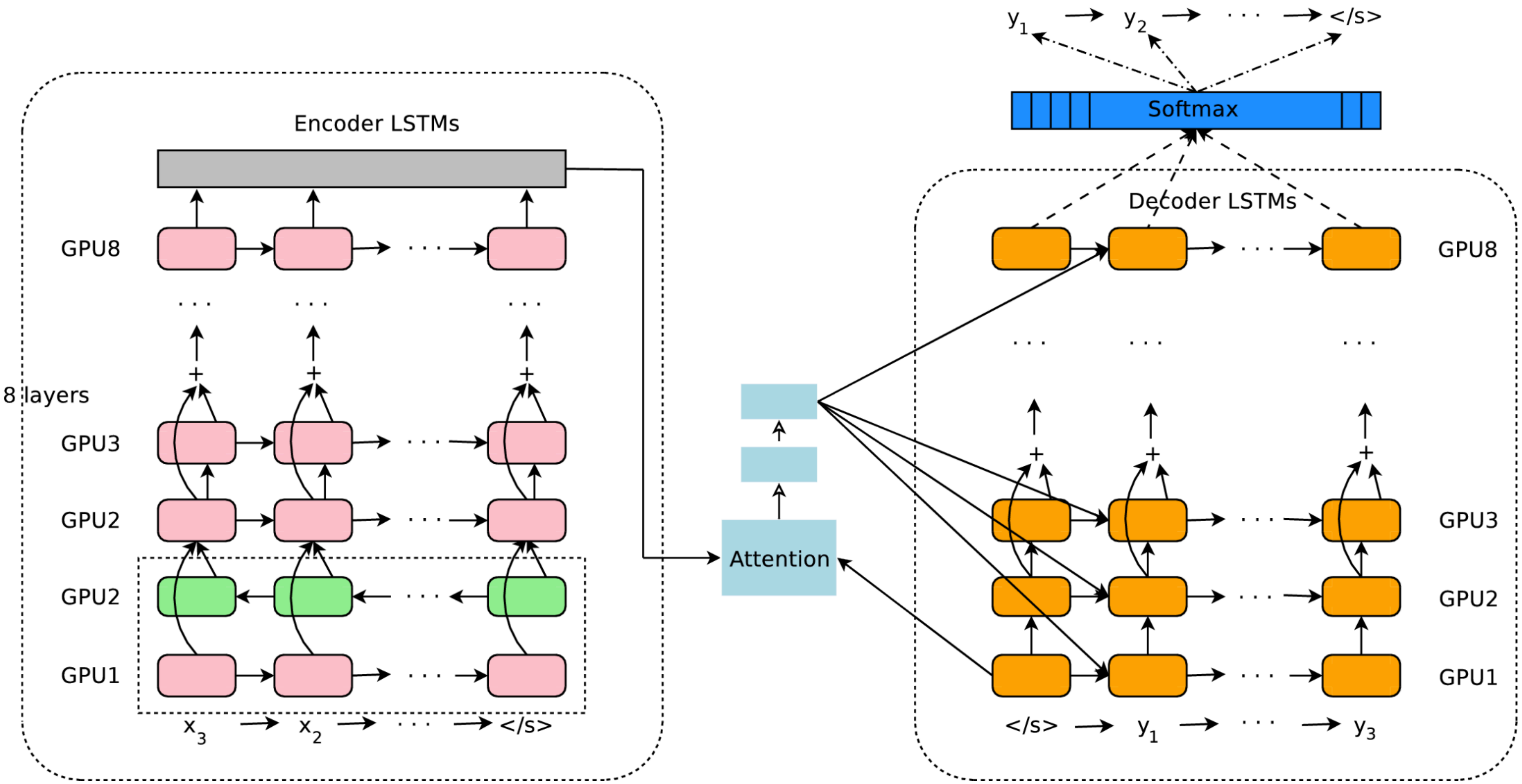
```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences
- Merge the most frequent pair of adjacent characters

- Final size = initial vocab + num merges. Often do 10k - 30k merges
- Most SOTA NMT systems use this on both source + target

Sennrich et al. (2016)

Google's NMT System



- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k

Wu et al. (2016)

Google's NMT System

Wu et al. (2016)

Google’s NMT System

English-French:

Google’s phrase-based system: 37.0 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU

Google’s 32k word pieces: 38.95 BLEU

Wu et al. (2016)

Google’s NMT System

English-French:

Google’s phrase-based system: 37.0 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU

Google’s 32k word pieces: 38.95 BLEU

English-German:

Google’s phrase-based system: 20.7 BLEU

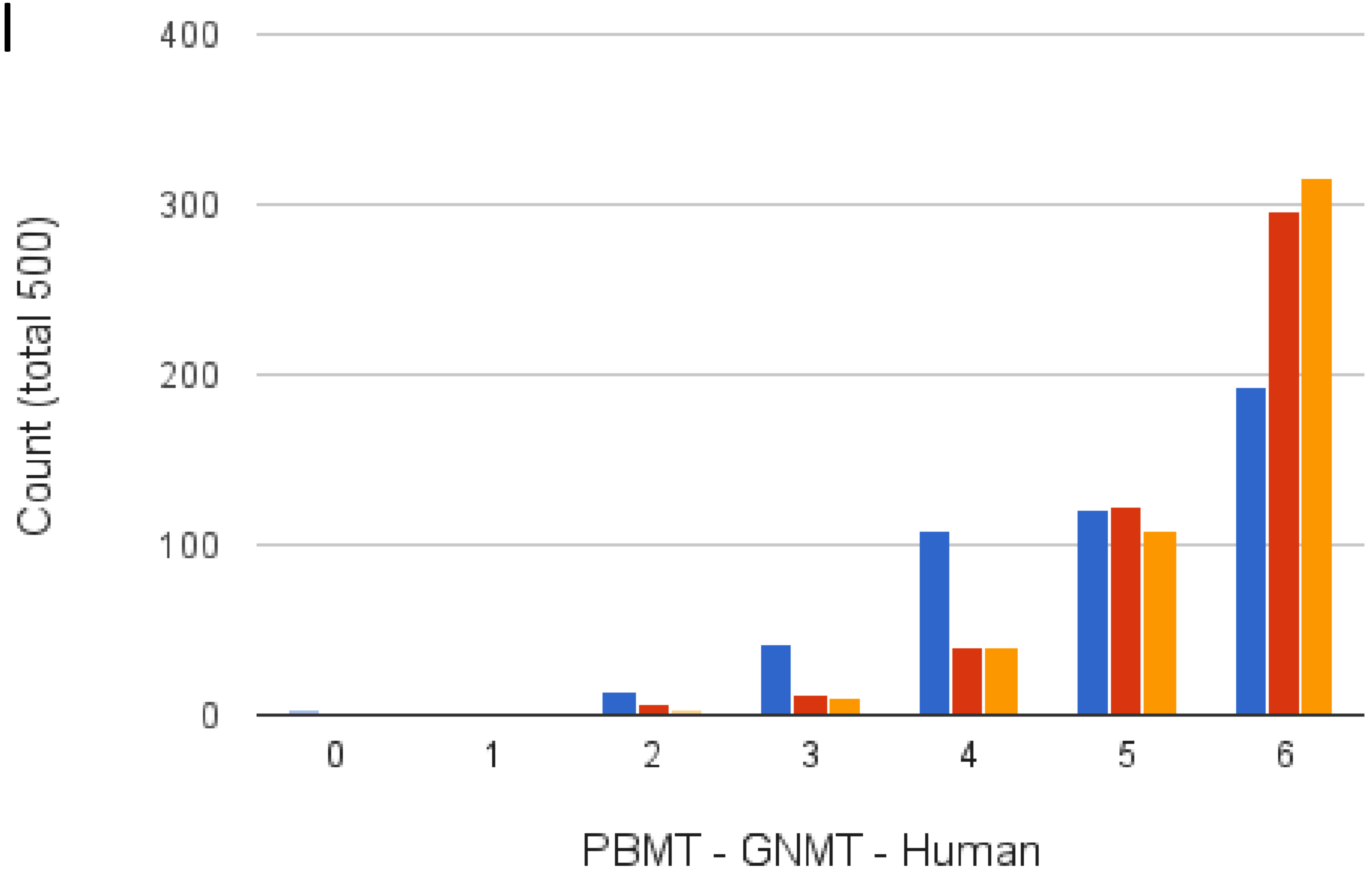
Luong+ (2015) seq2seq ensemble with rare word handling: 23.0 BLEU

Google’s 32k word pieces: 24.2 BLEU

Wu et al. (2016)

Human Evaluation (En-Es)

- Similar to human-level performance *on English-Spanish*



Wu et al. (2016)

Google's NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Wu et al. (2016)

Google's NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT
but not in PBMT

Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

“sled”

Wu et al. (2016)

Google's NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

"sled"

Wu et al. (2016)

Google's NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

"sled"

"walker"

Wu et al. (2016)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T’ to train a language model. Can neural MT do the same?

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?
- Approach 1: force the system to generate T' as targets from null inputs

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?
- Approach 1: force the system to generate T' as targets from null inputs

s_1, t_1

s_2, t_2

…

[null], t'_1

[null], t'_2

…

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?

- Approach 1: force the system to generate T' as targets from null inputs

- Approach 2: generate synthetic sources with a T->S machine translation system (backtranslation)

s_1, t_1

s_2, t_2

...

[null], t'_1

[null], t'_2

...

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?

- Approach 1: force the system to generate T' as targets from null inputs

s_1, t_1
s_2, t_2
…
[null], t'_1
[null], t'_2
…

- Approach 2: generate synthetic sources with a T->S machine translation system (backtranslation)

s_1, t_1
s_2, t_2
…
$MT(t'_1), t'_1$
$MT(t'_2), t'_2$
…

Sennrich et al. (2015)

Backtranslation

name	training data	instances	BLEU tst2011	tst2012	tst2013	tst2014
baseline (Gülçehre et al., 2015)			18.4	18.8	19.9	18.7
deep fusion (Gülçehre et al., 2015)			20.2	20.2	21.3	**20.6**
baseline	parallel	7.2m	18.6	18.2	18.4	18.3
$\text{parallel}_{\text{synth}}$	$\text{parallel/parallel}_{\text{synth}}$	6m/6m	19.9	20.4	20.1	20.0
$\text{Gigaword}_{\text{mono}}$	$\text{parallel/Gigaword}_{\text{mono}}$	7.6m/7.6m	18.8	19.6	19.4	18.2
$\text{Gigaword}_{\text{synth}}$	$\text{parallel/Gigaword}_{\text{synth}}$	8.4m/8.4m	**21.2**	**21.1**	**21.8**	20.4

- Gigaword: large monolingual English corpus
- $\text{parallel}_{\text{synth}}$: backtranslate training data; makes additional noisy source sentences which could be useful

Sennrich et al. (2015)

Dilated CNNs for MT

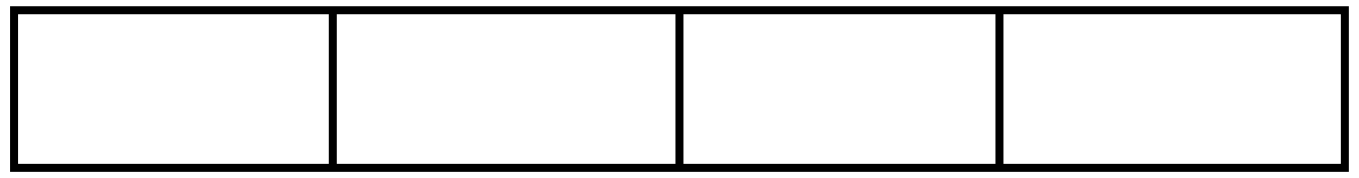
Dilated Convolutions

- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token

Strubell et al. (2017)

Dilated Convolutions

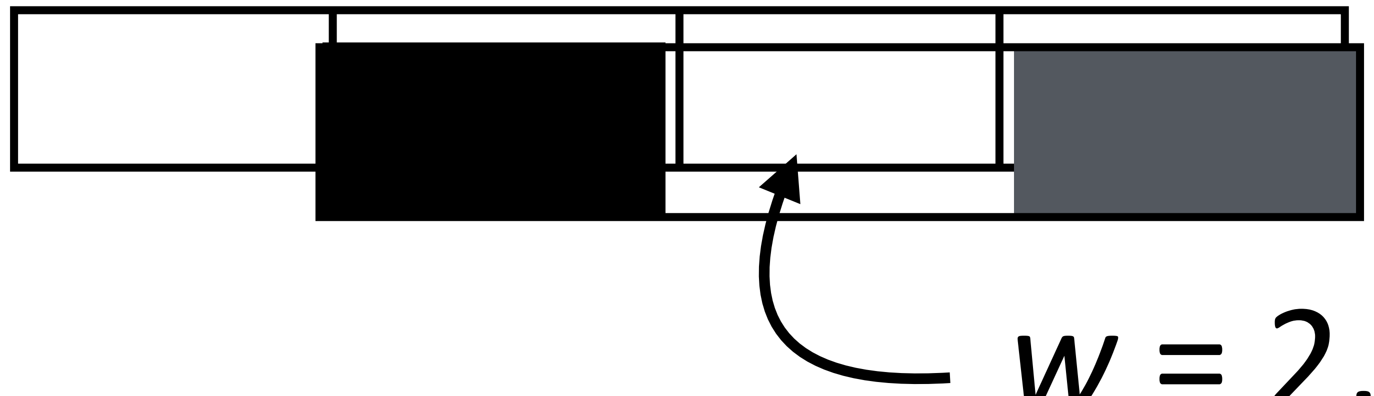
- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token



Strubell et al. (2017)

Dilated Convolutions

- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token

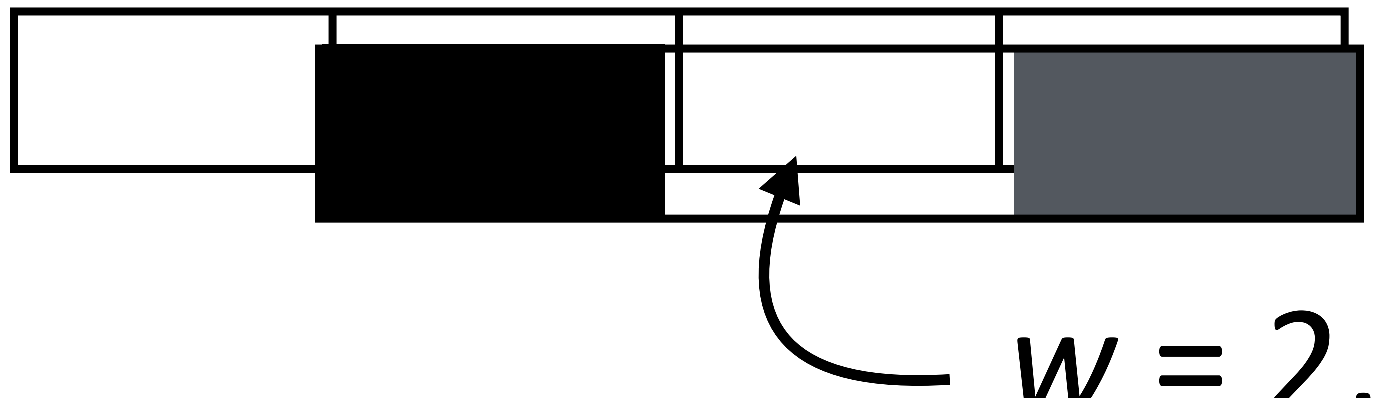


w = 2, d =2: gap in the filter

Strubell et al. (2017)

Dilated Convolutions

- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token



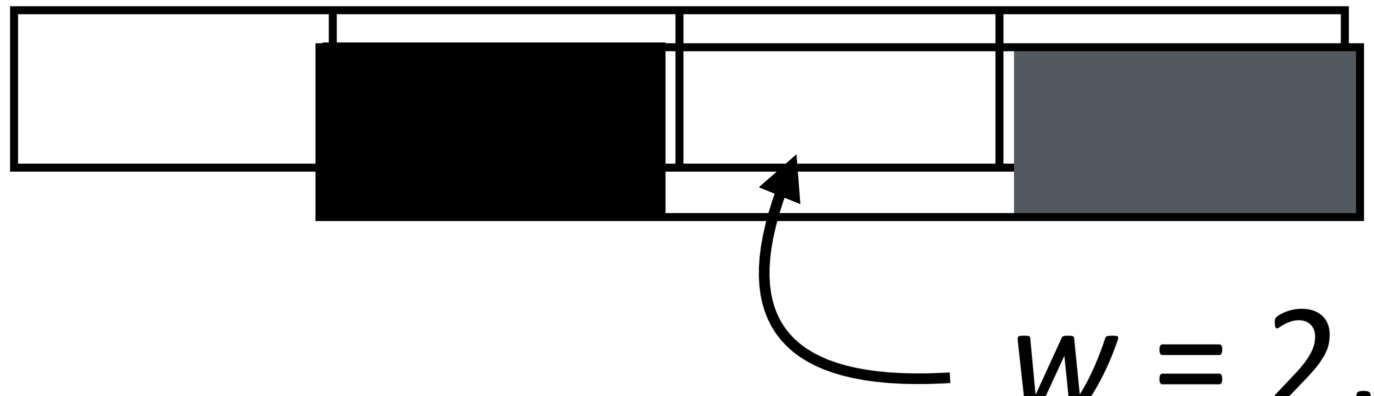
w = 2, d =2: gap in the filter

- Can chain successive dilated convolutions together to get a wide receptive field (see a lot of the sentence)

Strubell et al. (2017)

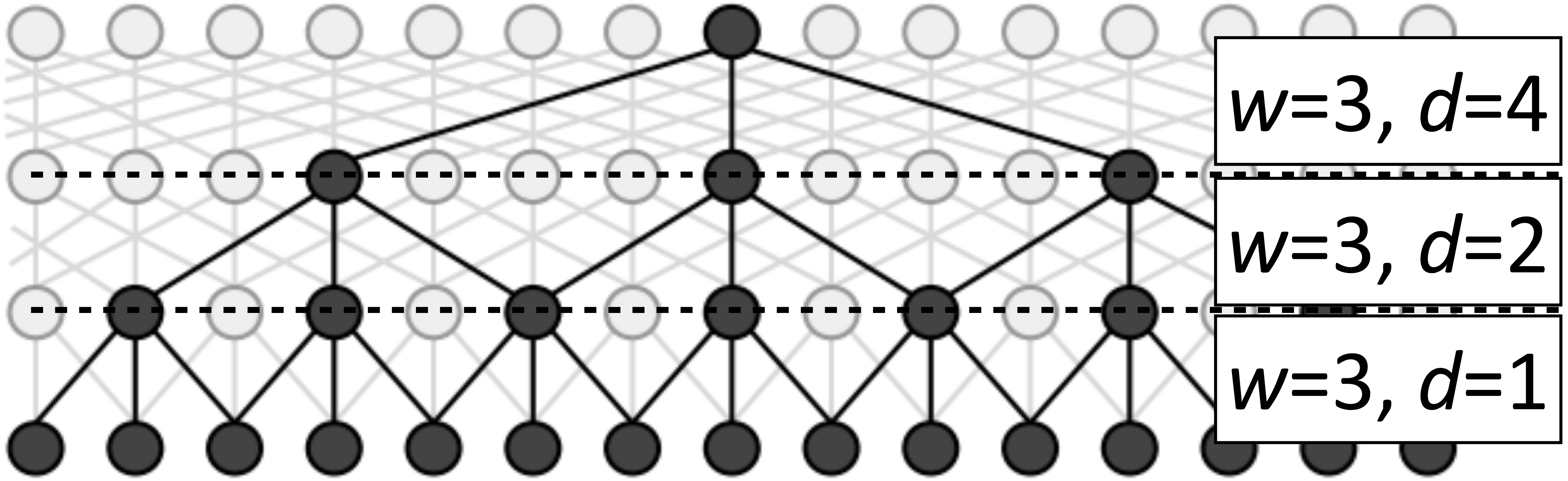
Dilated Convolutions

- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token



w = 2, *d* =2: gap in the filter

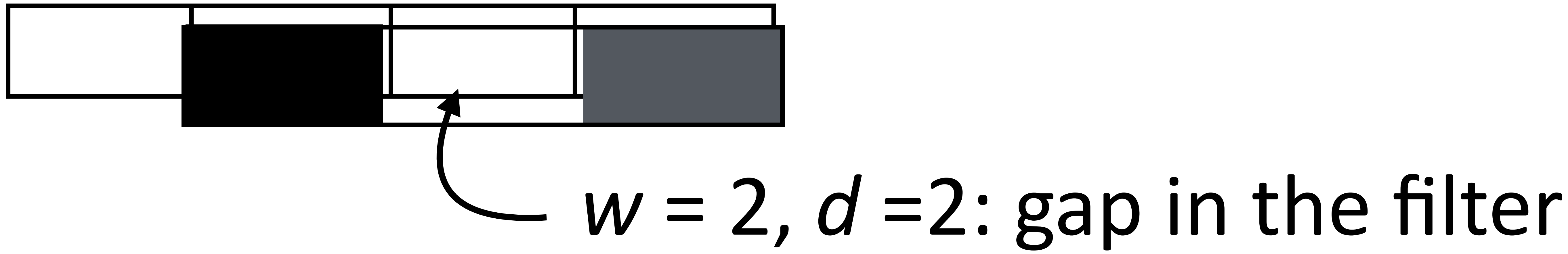
- Can chain successive dilated convolutions together to get a wide receptive field (see a lot of the sentence)



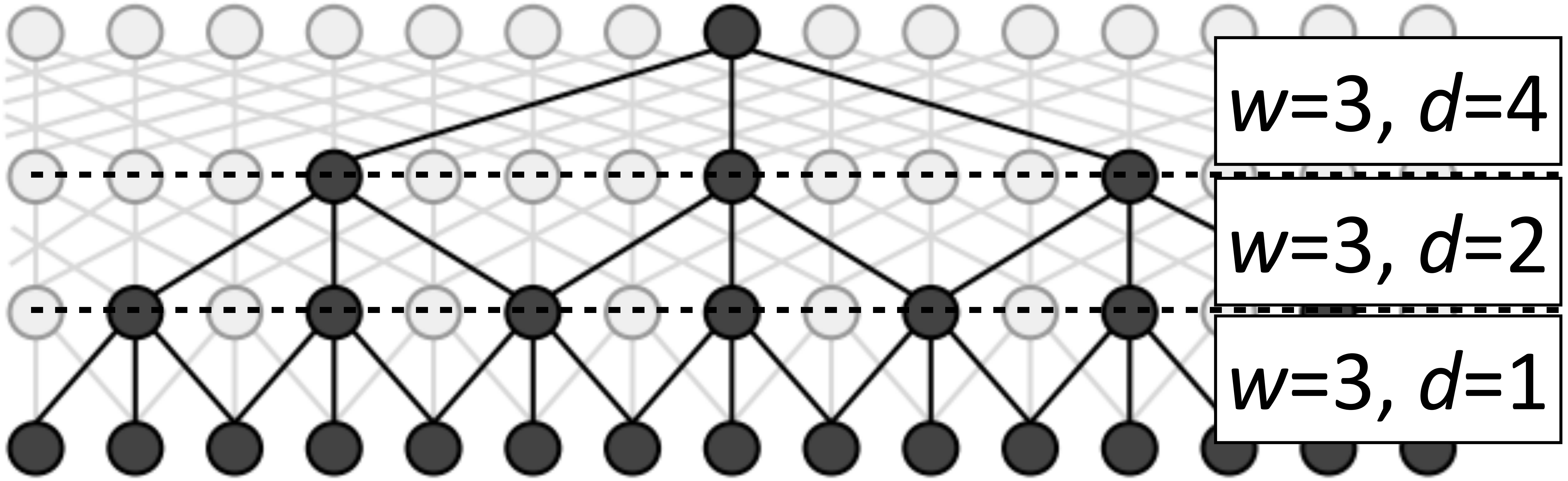
Strubell et al. (2017)

Dilated Convolutions

- Standard convolution: looks at every token under the filter
- Dilated convolution with gap *d*: looks at every *d*th token



- Can chain successive dilated convolutions together to get a wide receptive field (see a lot of the sentence)

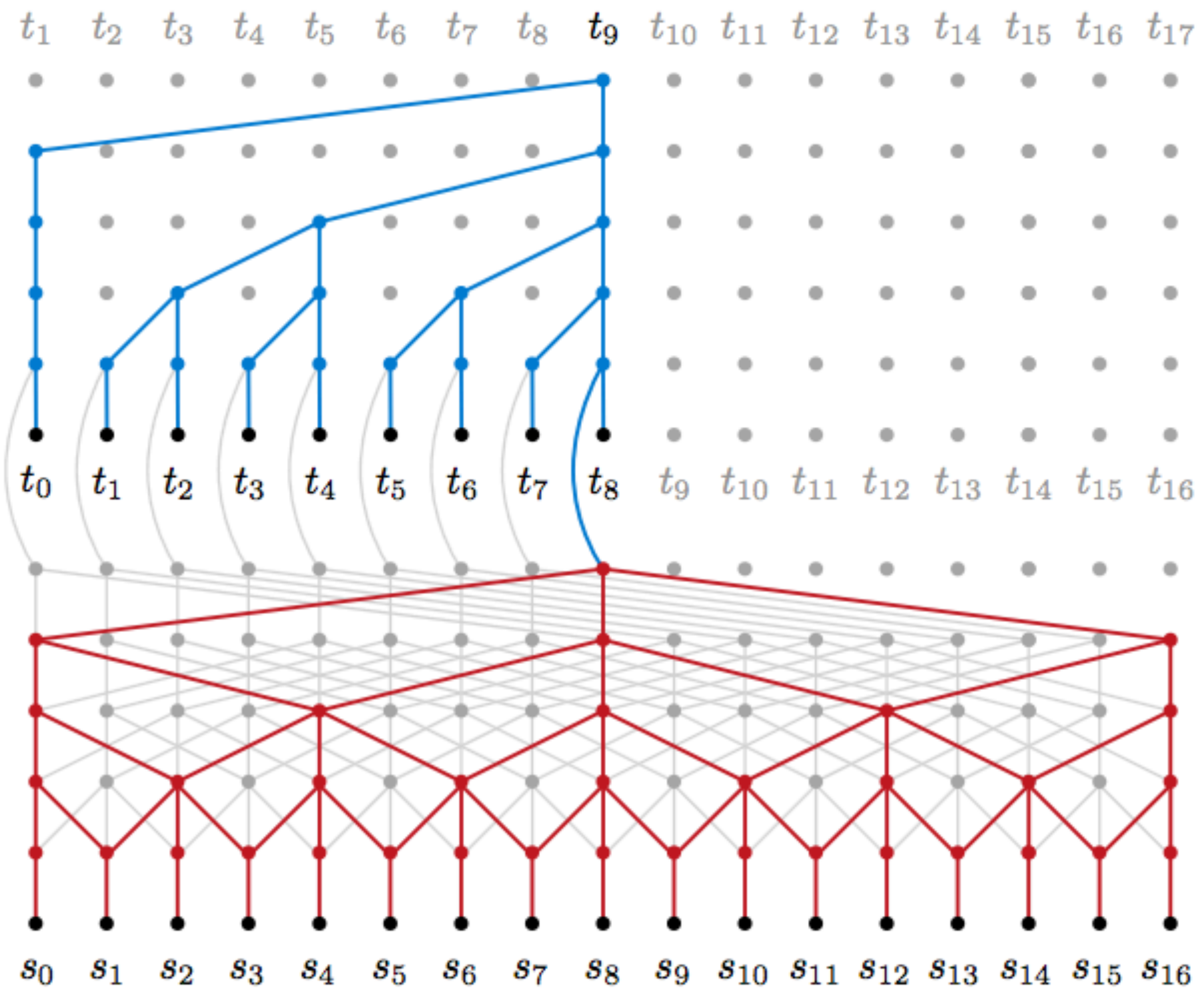


- Top nodes see lots of the sentence, but with different processing

Strubell et al. (2017)

CNNs for Machine Translation

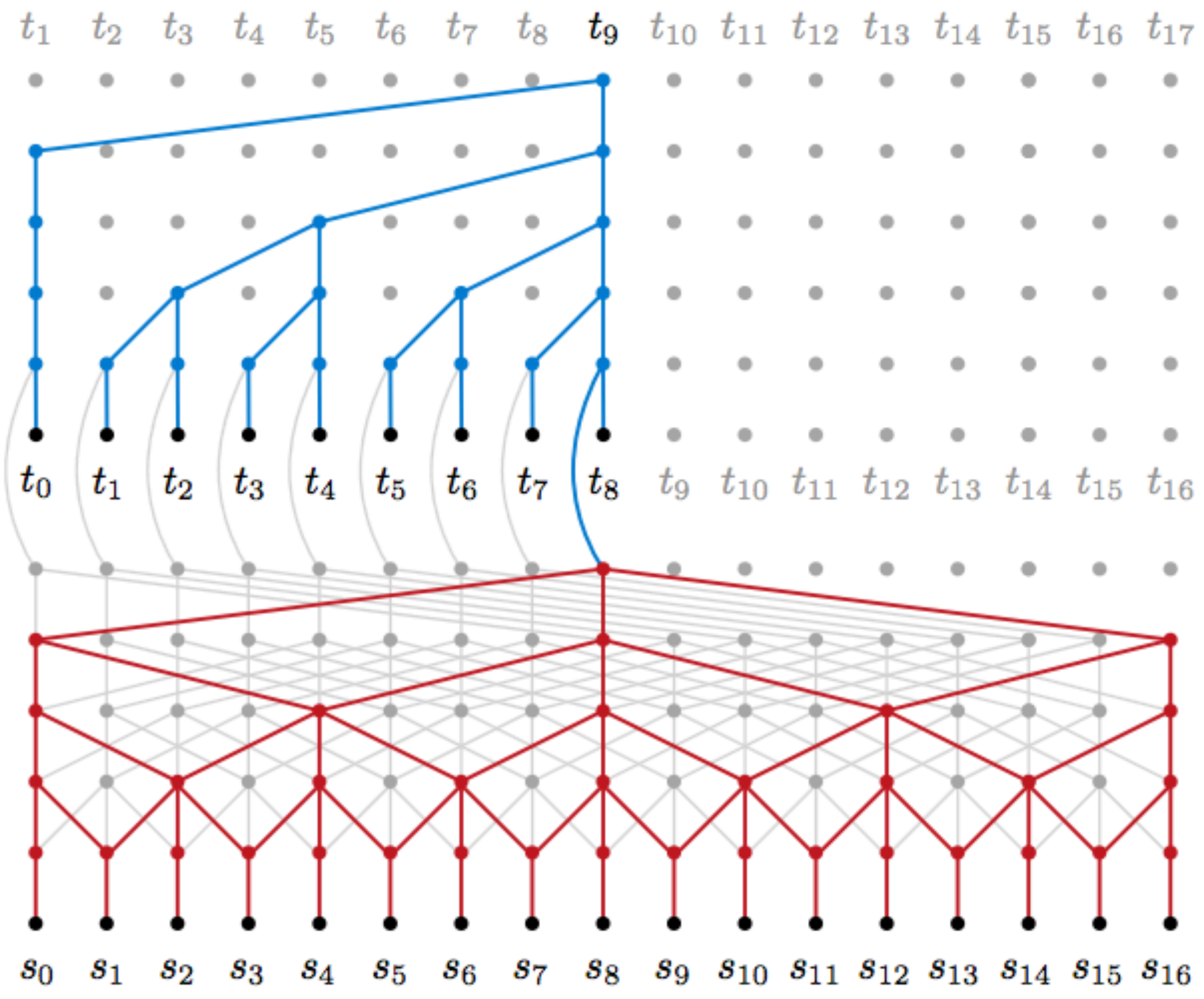
- “ByteNet”: operates over characters (bytes)



Kalchbrenner et al. (2016)

CNNs for Machine Translation

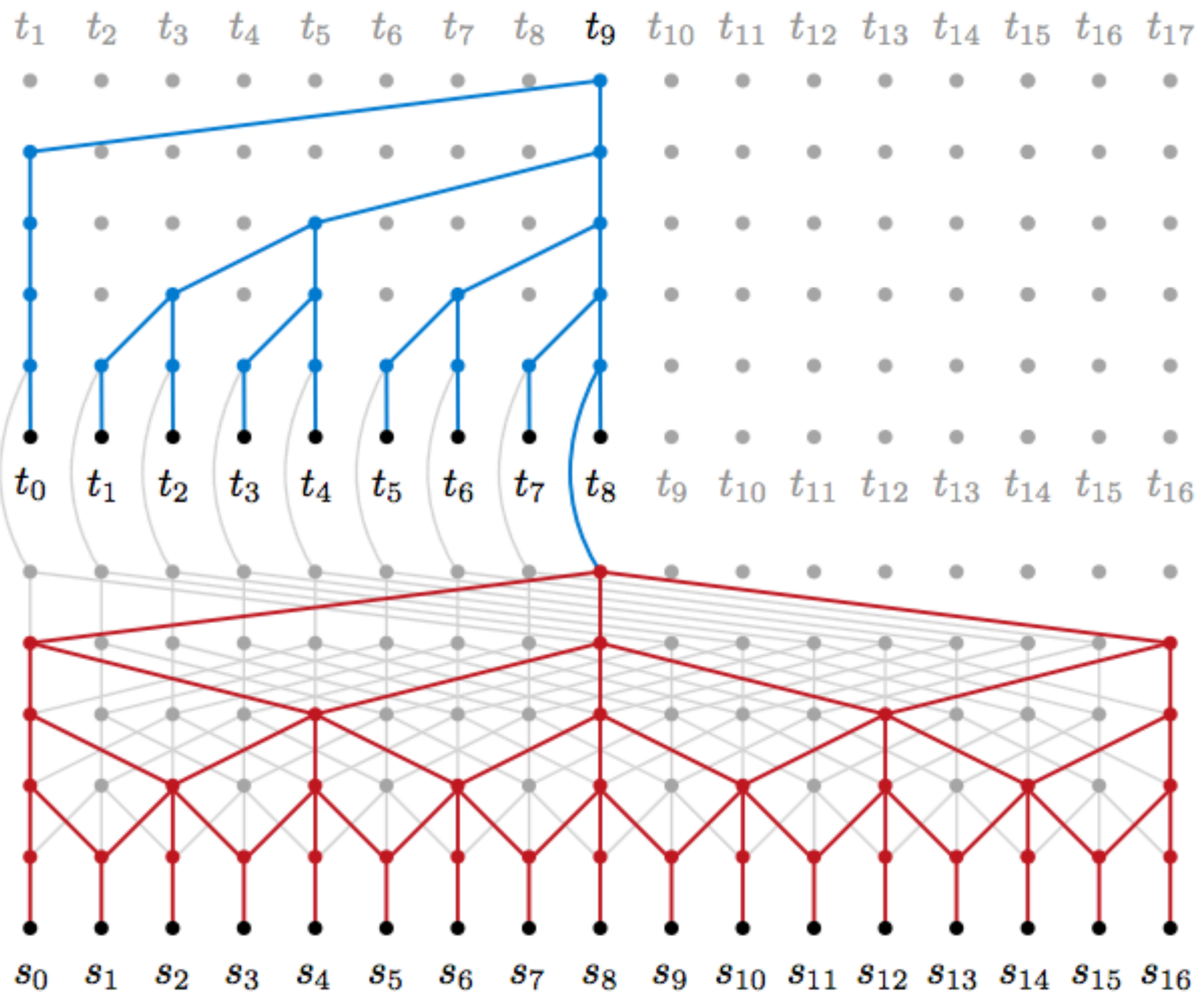
- “ByteNet”: operates over characters (bytes)
- Encode source sequence w/dilated convolutions



Kalchbrenner et al. (2016)

CNNs for Machine Translation

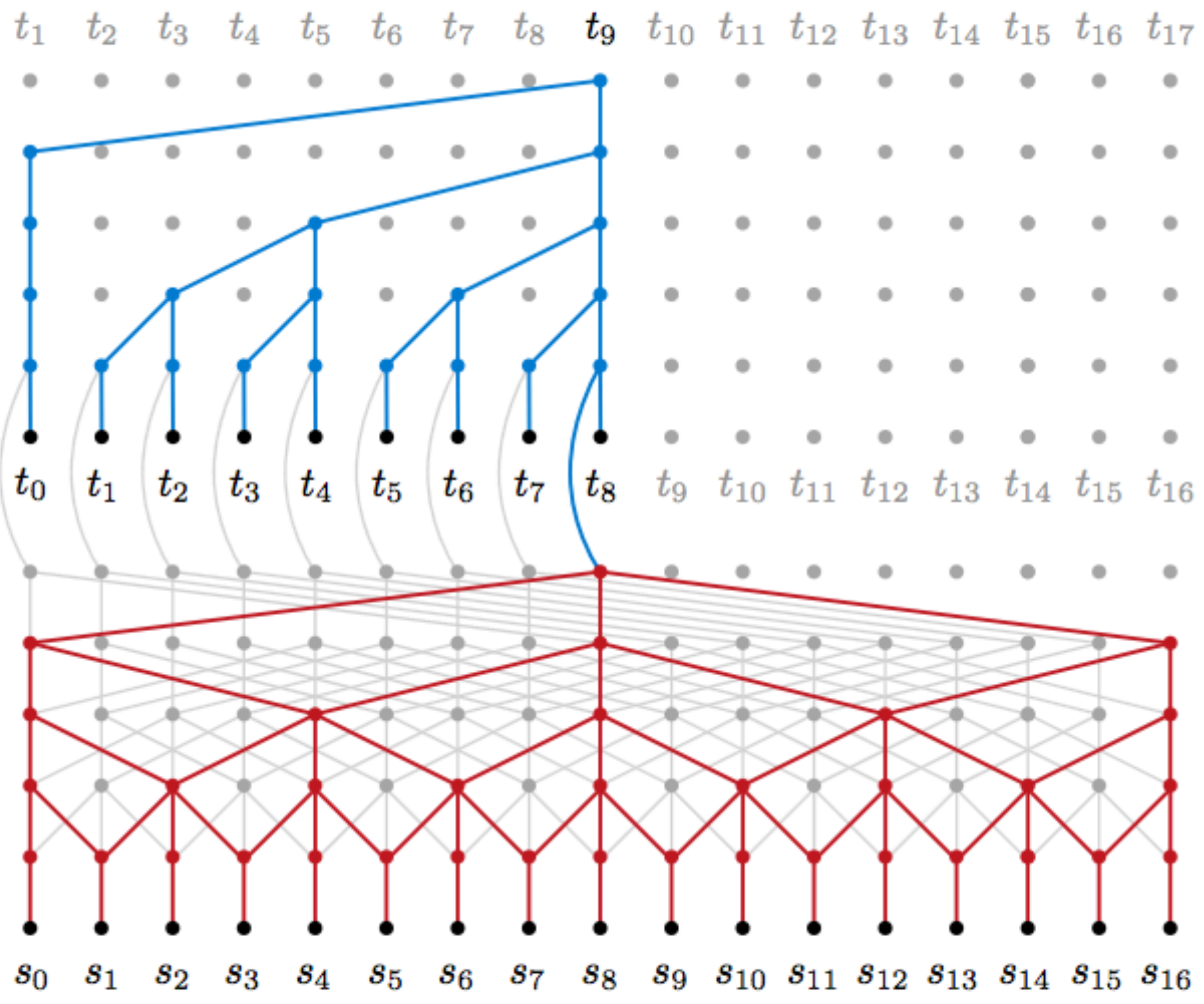
- "ByteNet": operates over characters (bytes)
- Encode source sequence w/dilated convolutions
- Predict *n*th target character by looking at the *n*th position in the source and a dilated convolution over the *n*-1 target tokens so far



Kalchbrenner et al. (2016)

CNNs for Machine Translation

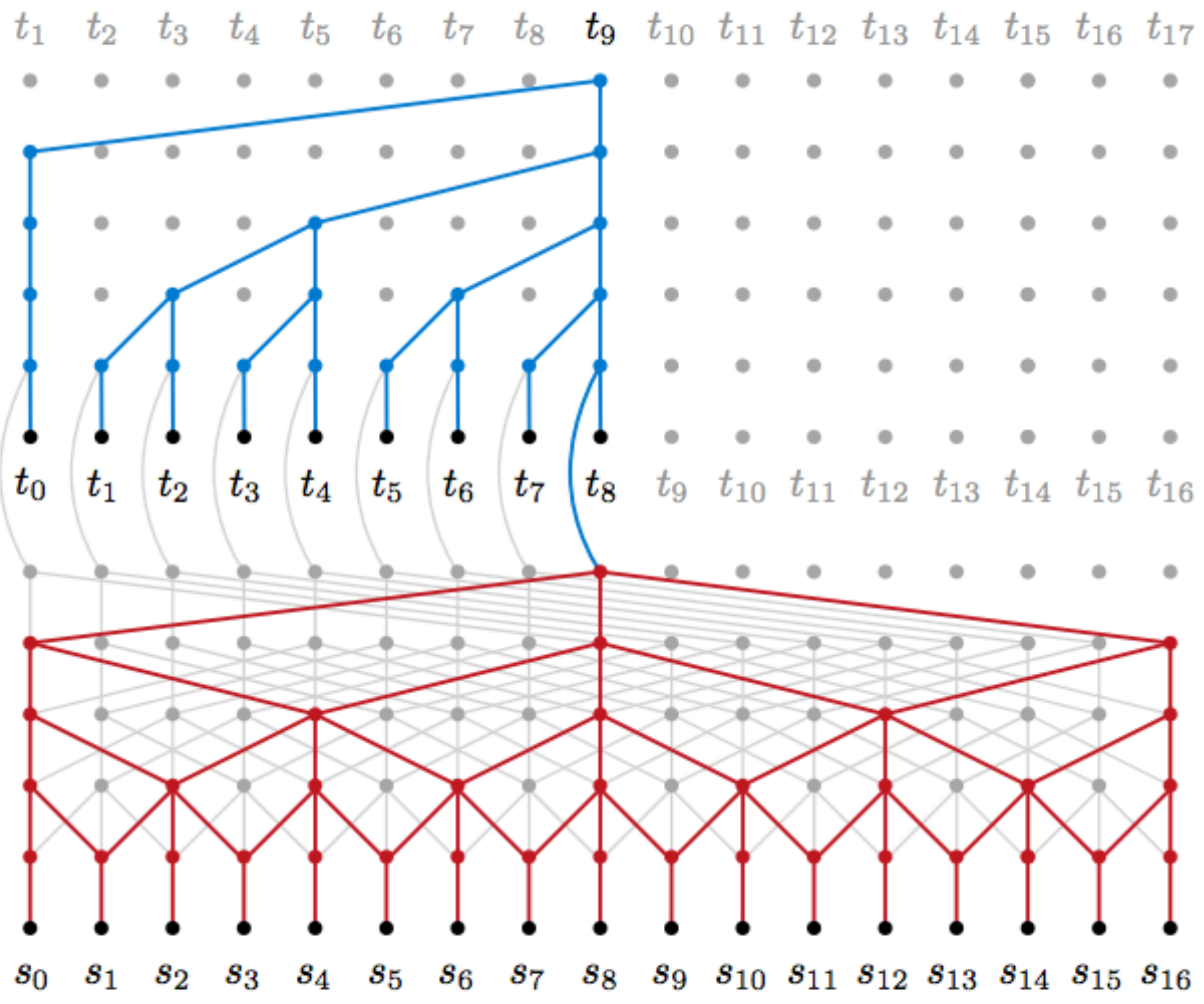
- "ByteNet": operates over characters (bytes)
- Encode source sequence w/dilated convolutions
- Predict *n*th target character by looking at the *n*th position in the source and a dilated convolution over the *n*-1 target tokens so far
- To deal with divergent lengths, t_n actually looks at $s_{n\alpha}$ where α is a heuristically-chosen parameter



Kalchbrenner et al. (2016)

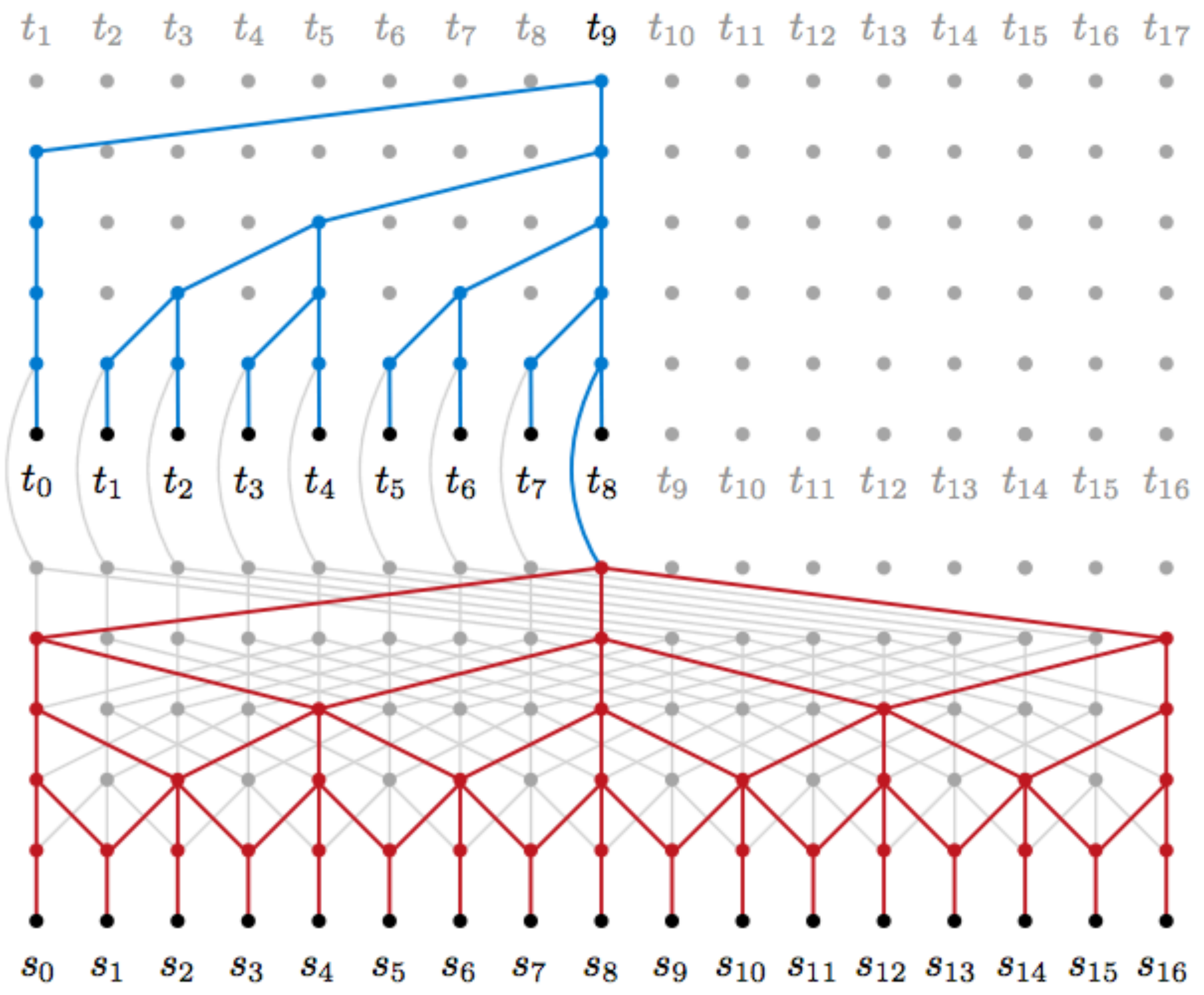
CNNs for Machine Translation

- "ByteNet": operates over characters (bytes)
- Encode source sequence w/dilated convolutions
- Predict *n*th target character by looking at the *n*th position in the source and a dilated convolution over the *n*-1 target tokens so far
- To deal with divergent lengths, t_n actually looks at $s_{n\alpha}$ where α is a heuristically-chosen parameter
- Assumes mostly monotonic translation



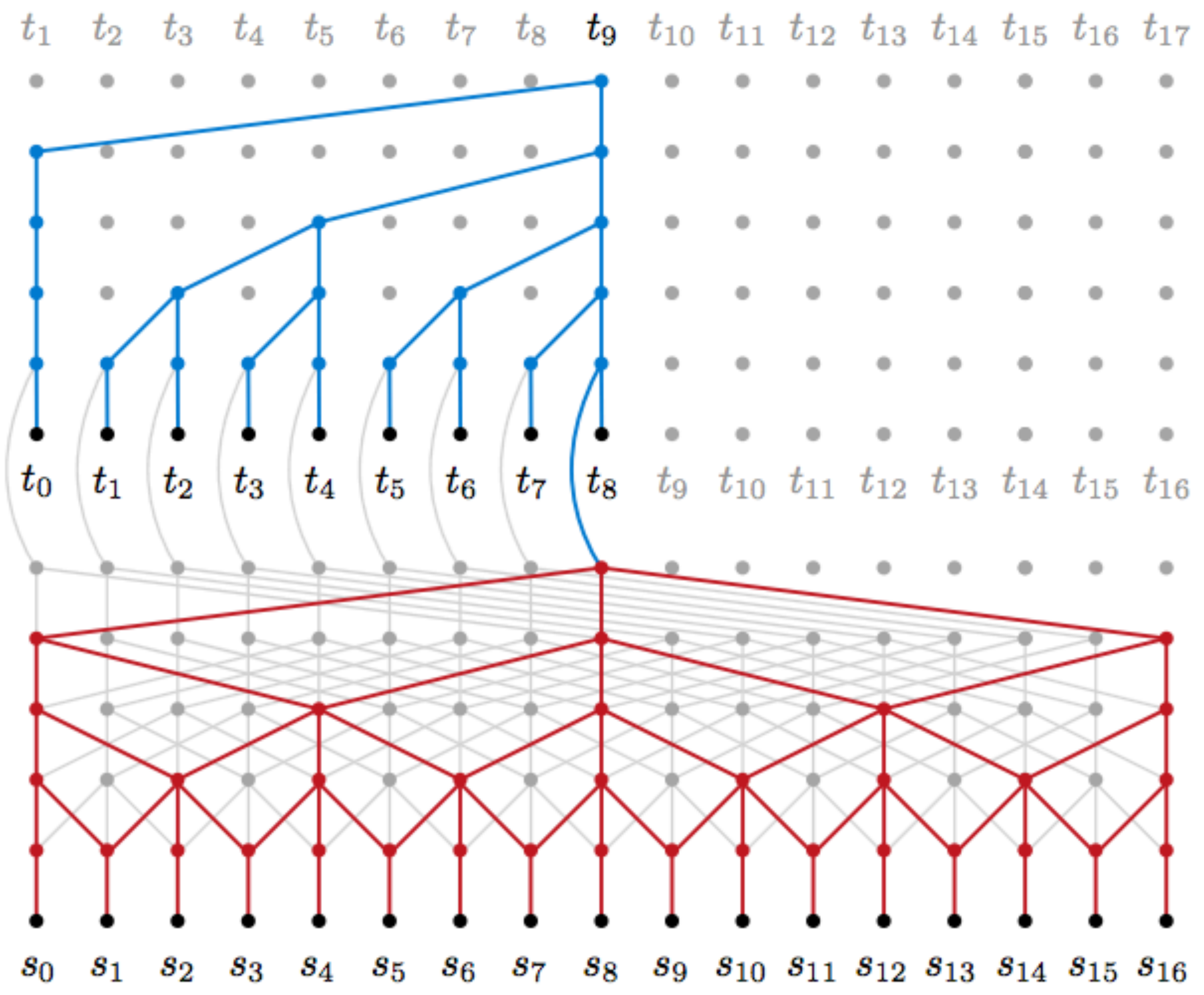
Kalchbrenner et al. (2016)

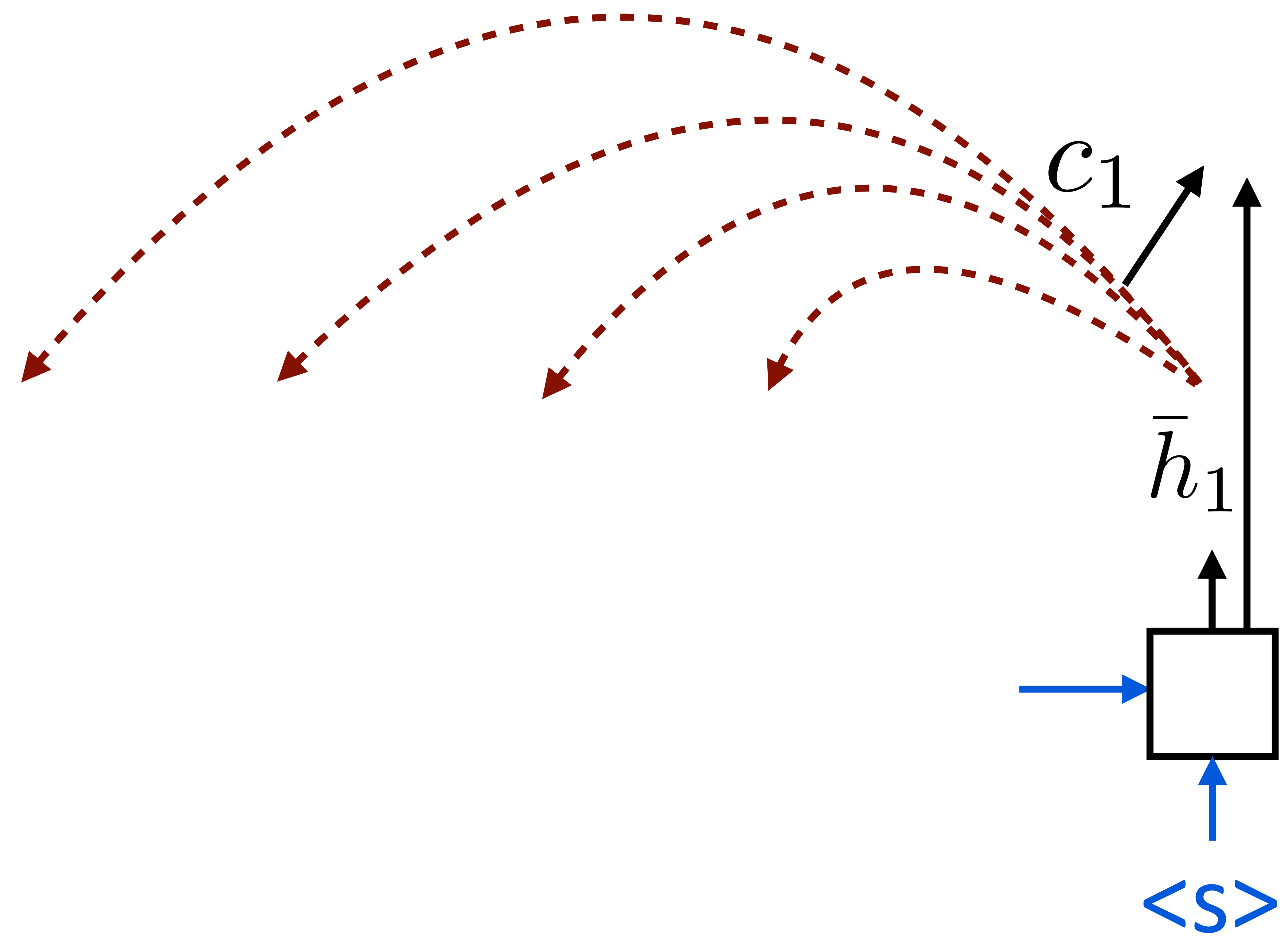
Compare: CNNs vs. LSTMs



Kalchbrenner et al. (2016)

Compare: CNNs vs. LSTMs

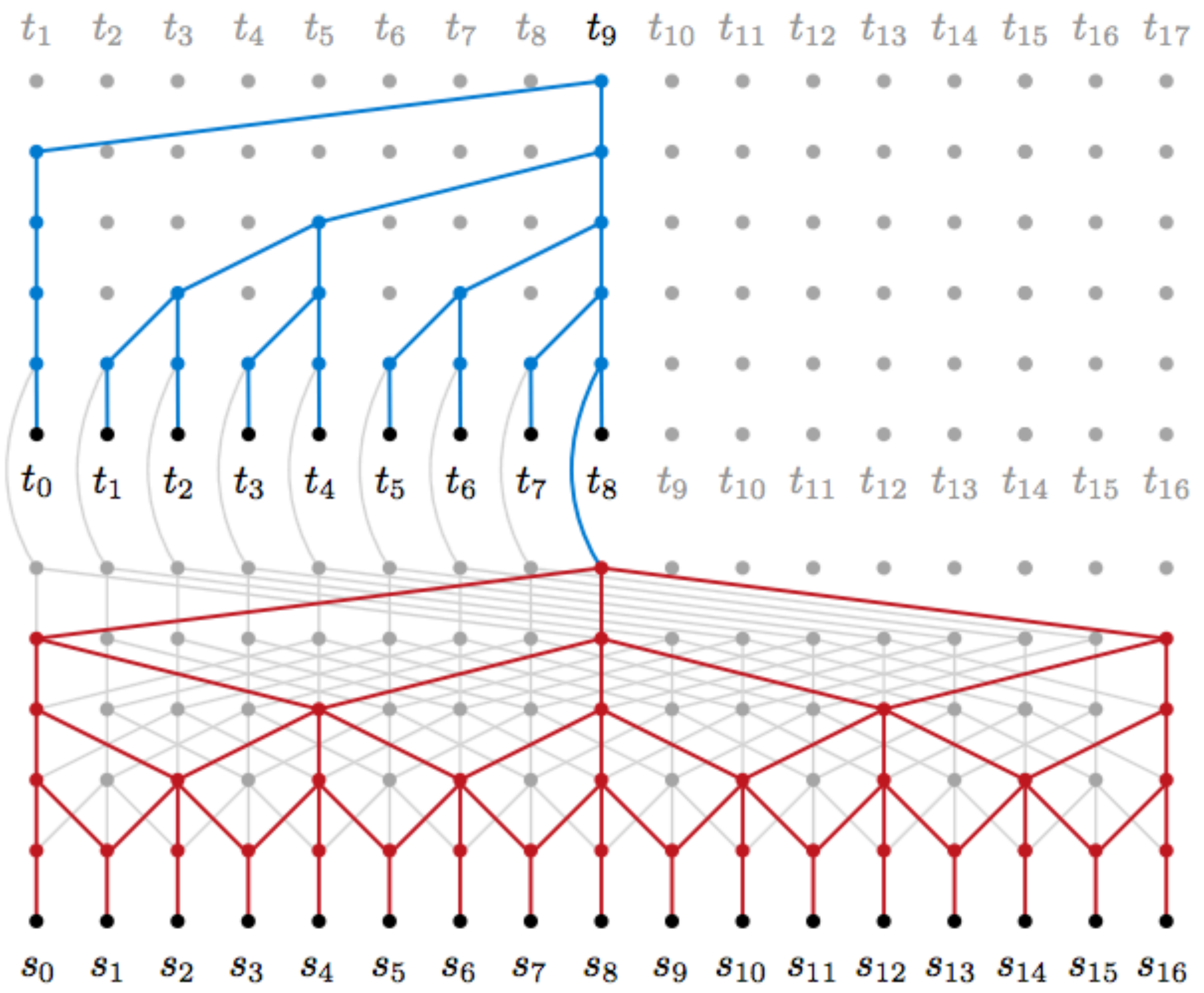


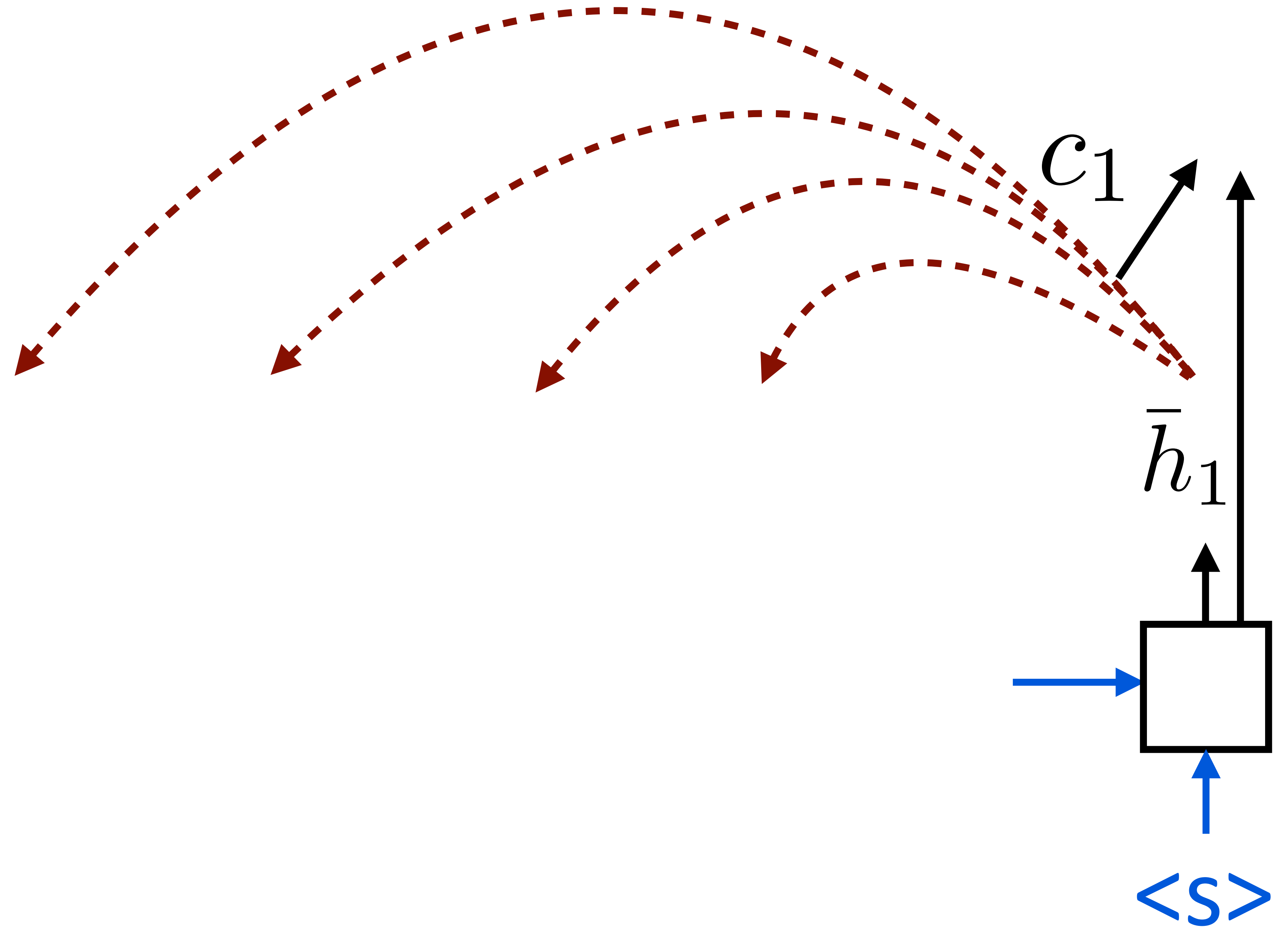


- LSTM: looks at previous word + hidden state, attention over input

Kalchbrenner et al. (2016)

Compare: CNNs vs. LSTMs

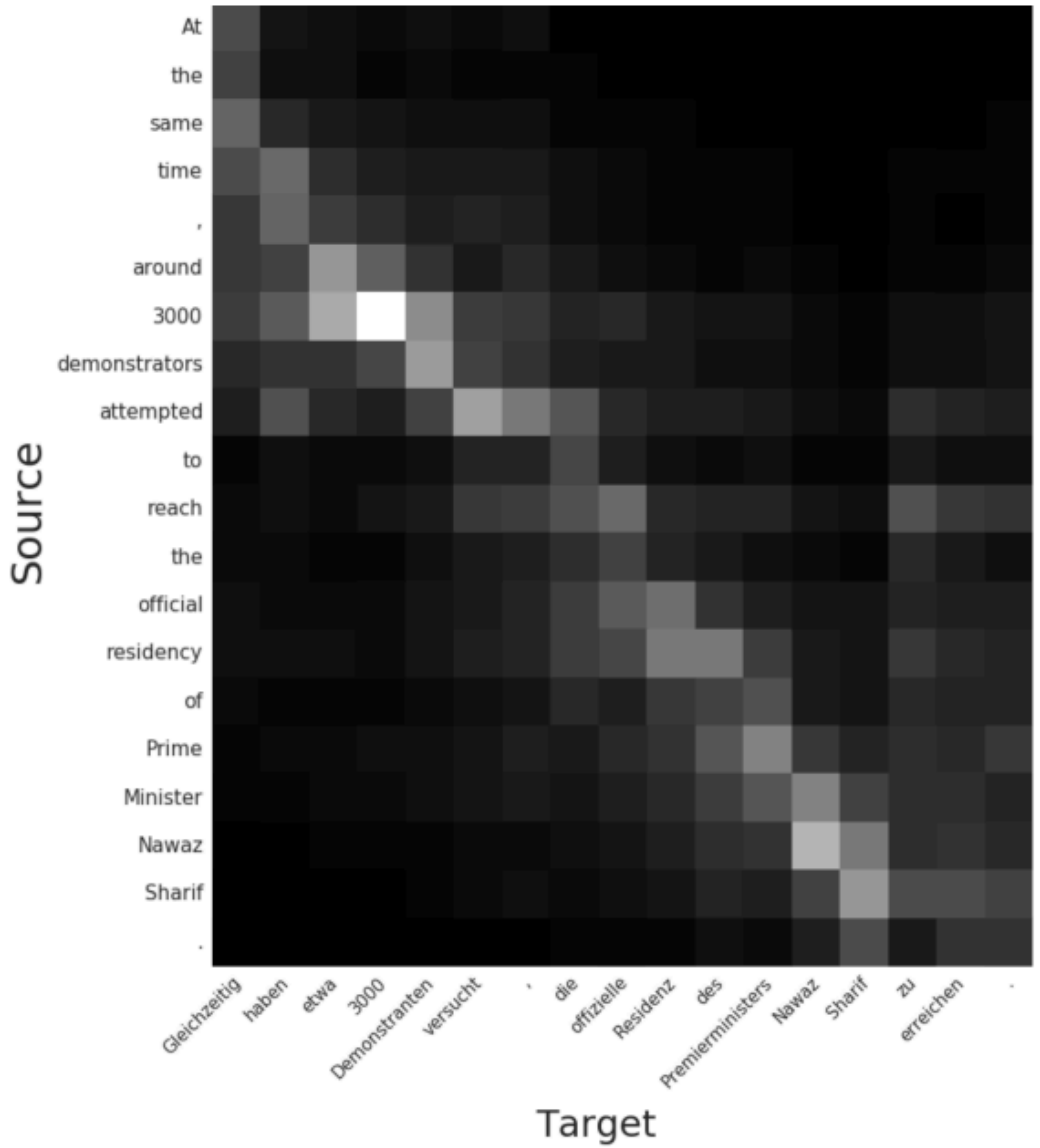




- CNN: source encoding at this position gives us "attention", target encoding gives us decoder context
- LSTM: looks at previous word + hidden state, attention over input

Kalchbrenner et al. (2016)

Attention from CNN

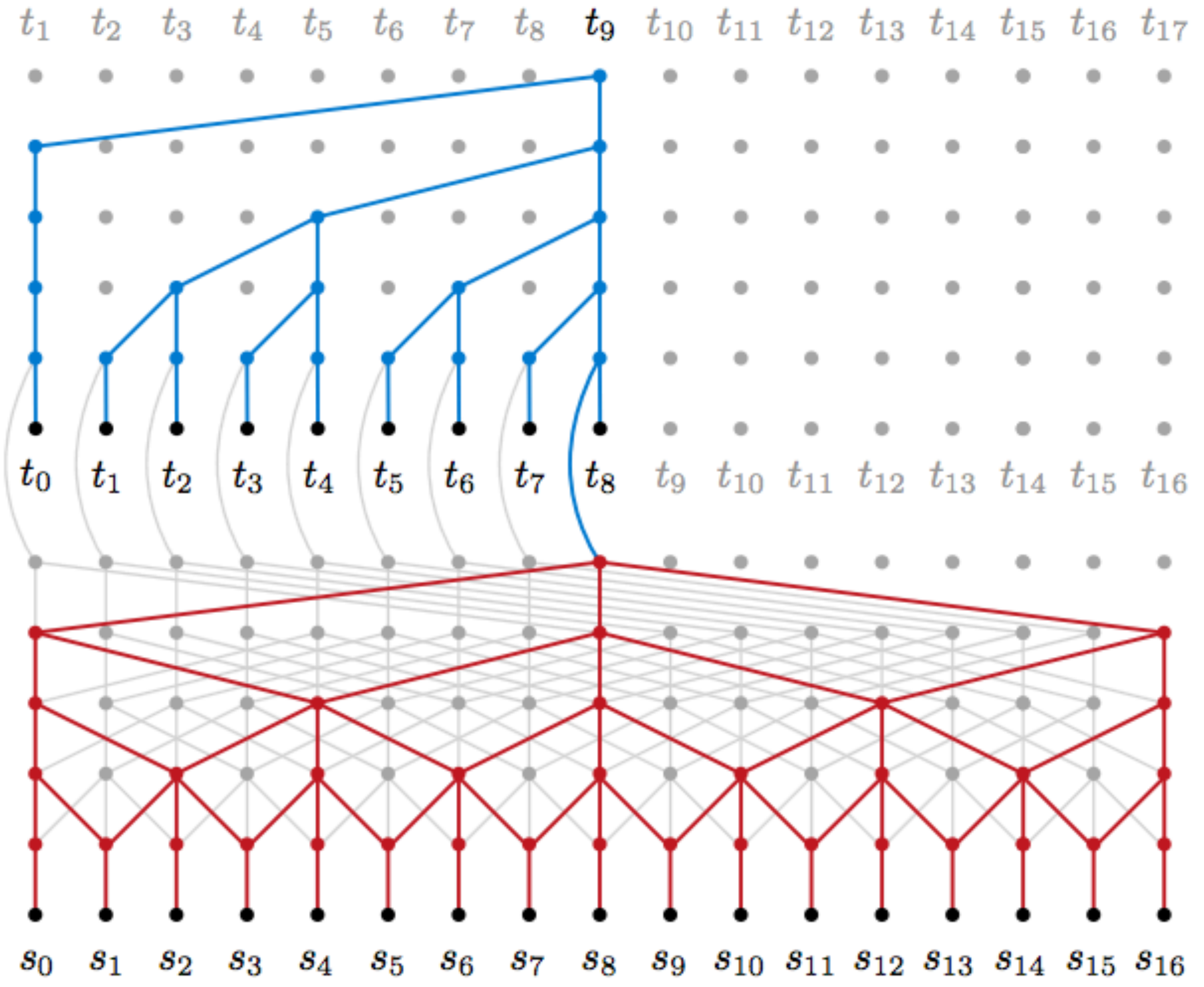


- Model is character-level, this visualization shows which words's characters impact the convolutional encoding the most
- Largely monotonic but does consult other information

Kalchbrenner et al. (2016)

Advantages of CNNs

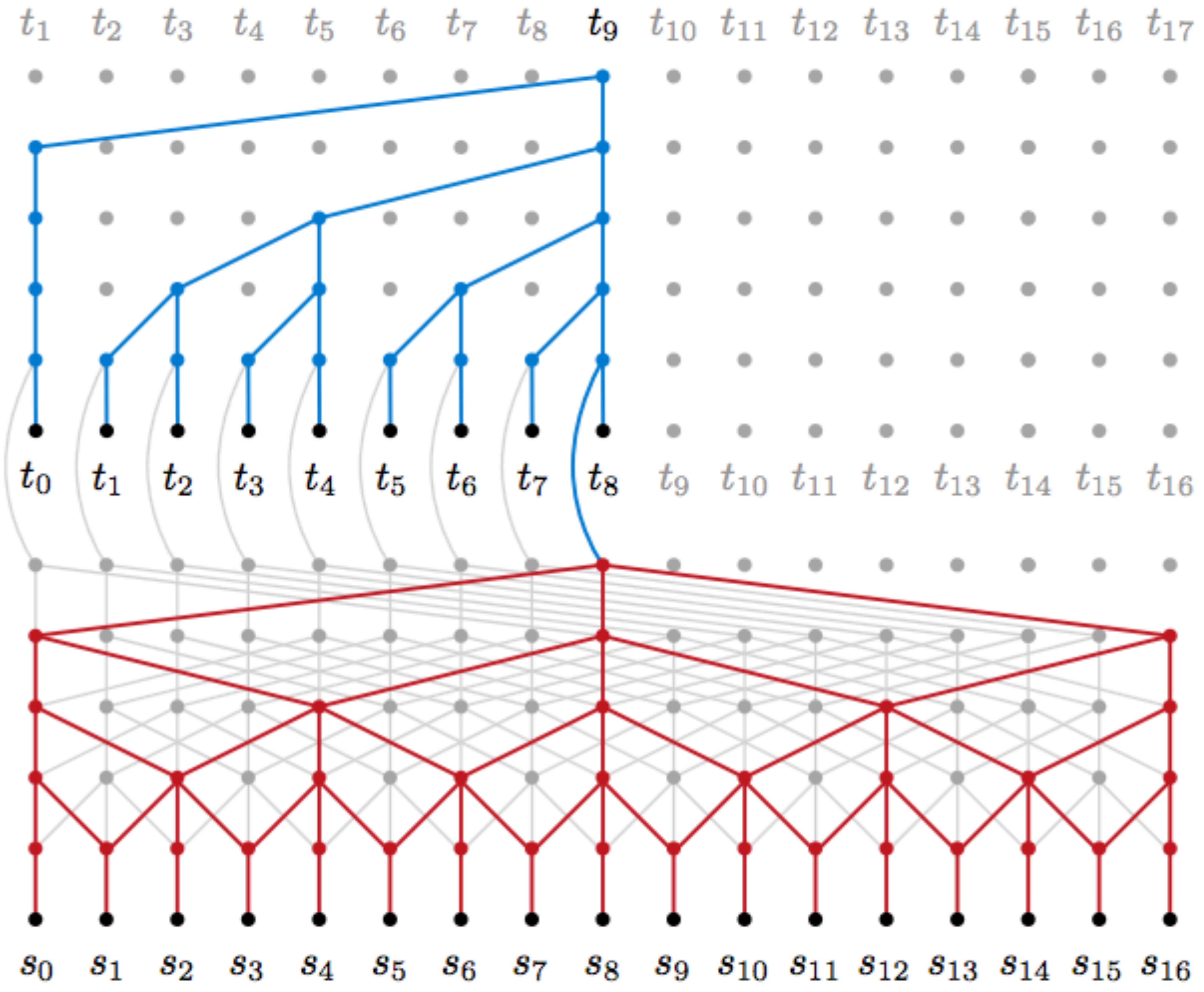
- LSTM with attention is quadratic: compute attention over the whole input for each decoded token



Kalchbrenner et al. (2016)

Advantages of CNNs

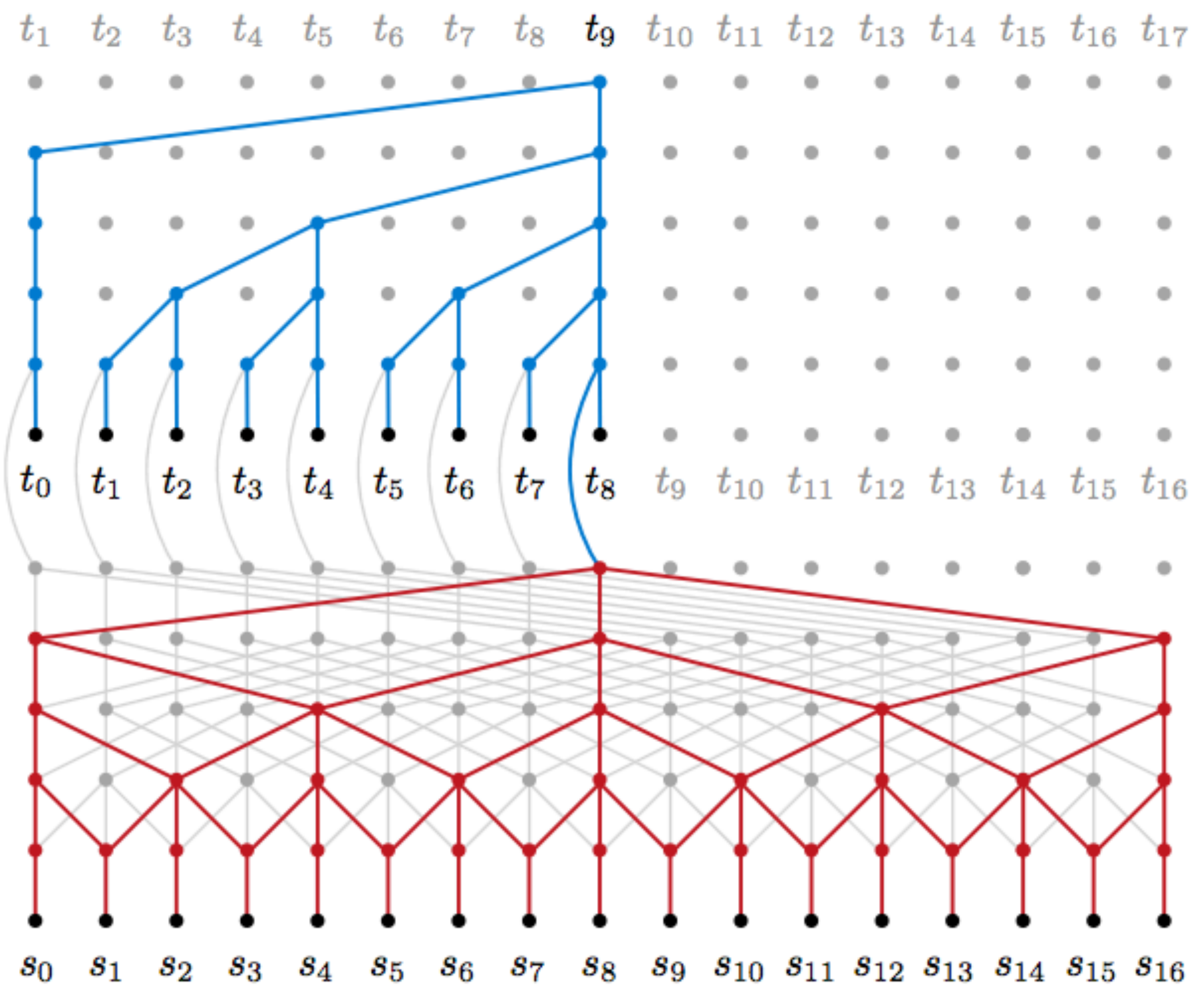
- LSTM with attention is quadratic: compute attention over the whole input for each decoded token
- CNN is linear!



Kalchbrenner et al. (2016)

Advantages of CNNs

- LSTM with attention is quadratic: compute attention over the whole input for each decoded token
- CNN is linear!
- CNN is shallower too in principle but the conv layers are very sophisticated (3 layers each)



Kalchbrenner et al. (2016)

English-German MT Results

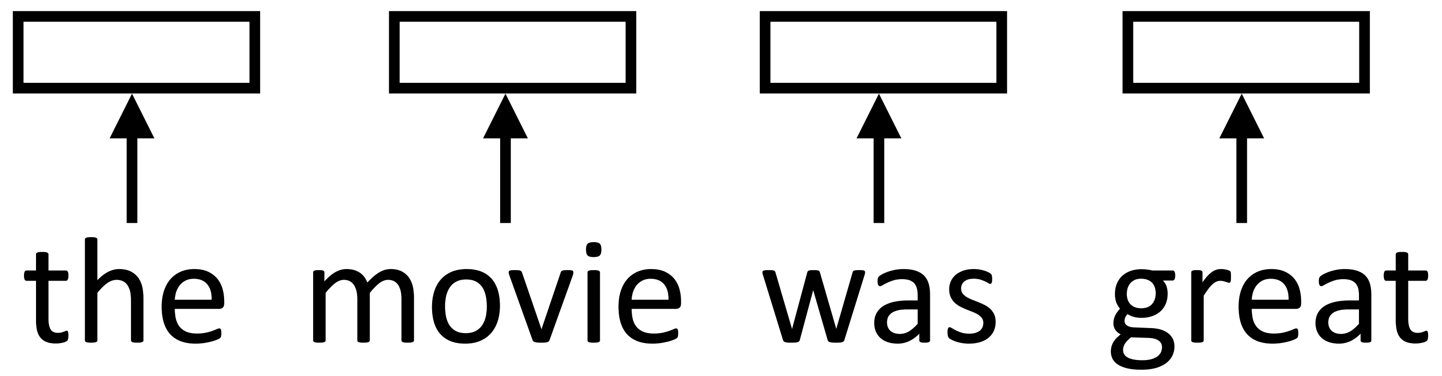
Model	Inputs	Outputs	WMT Test ’14
Phrase Based MT (Freitag et al., 2014; Williams et al., 2015)	phrases	phrases	20.7
RNN Enc-Dec (Luong et al., 2015)	words	words	11.3
Reverse RNN Enc-Dec (Luong et al., 2015)	words	words	14.0
RNN Enc-Dec Att (Zhou et al., 2016)	words	words	20.6
RNN Enc-Dec Att (Luong et al., 2015)	words	words	20.9
GNMT (RNN Enc-Dec Att) (Wu et al., 2016a)	word-pieces	word-pieces	**24.61**
RNN Enc-Dec Att (Chung et al., 2016b)	BPE	BPE	19.98
RNN Enc-Dec Att (Chung et al., 2016b)	BPE	char	21.33
GNMT (RNN Enc-Dec Att) (Wu et al., 2016a)	char	char	22.62
ByteNet	char	char	**23.75**

Kalchbrenner et al. (2016)

Transformers for MT

Self-Attention

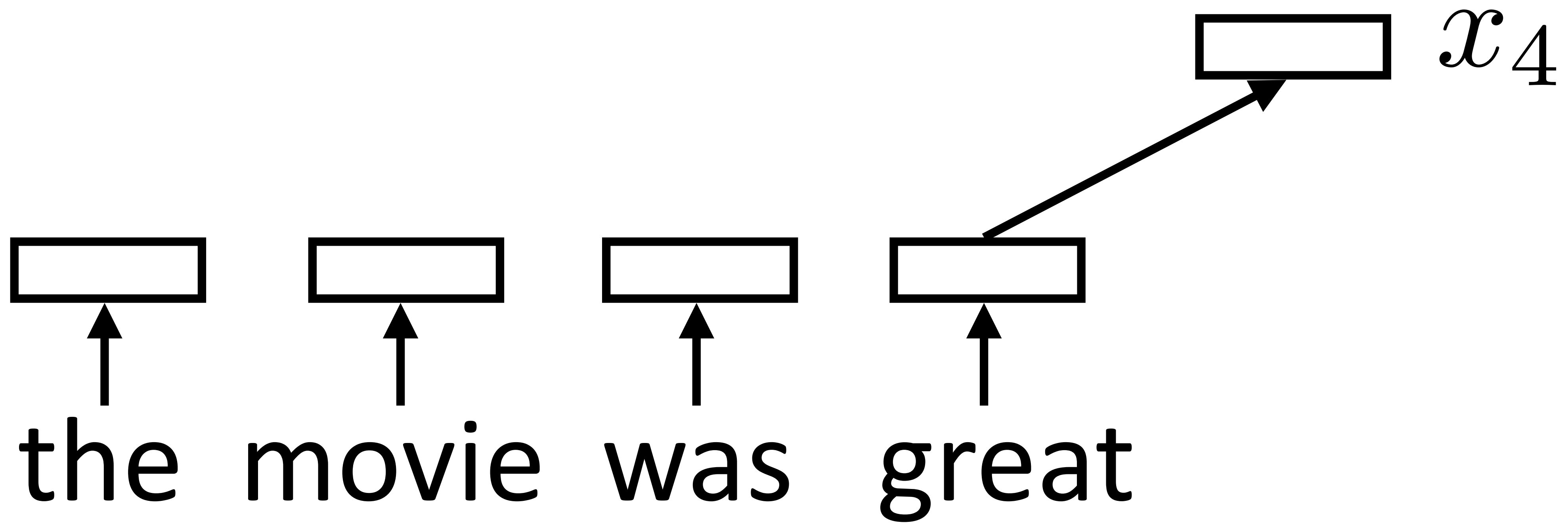
- Each word forms a "query" which then computes attention over each word



Vaswani et al. (2017)

Self-Attention

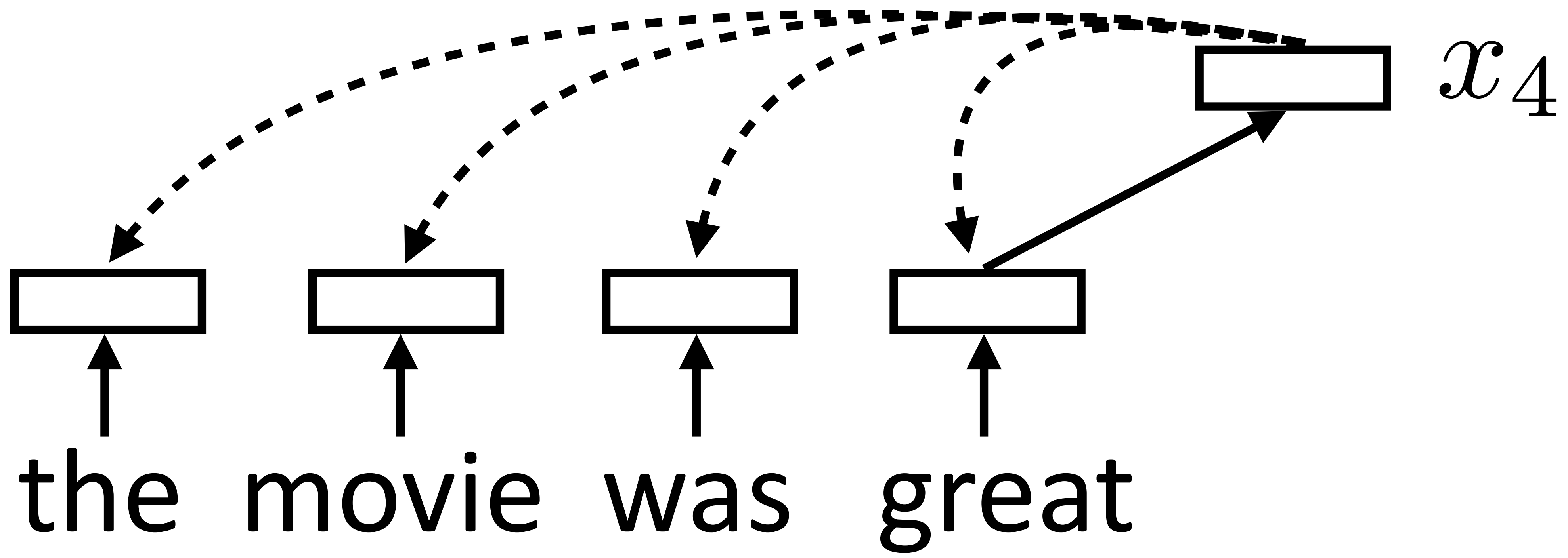
- Each word forms a “query” which then computes attention over each word



Vaswani et al. (2017)

Self-Attention

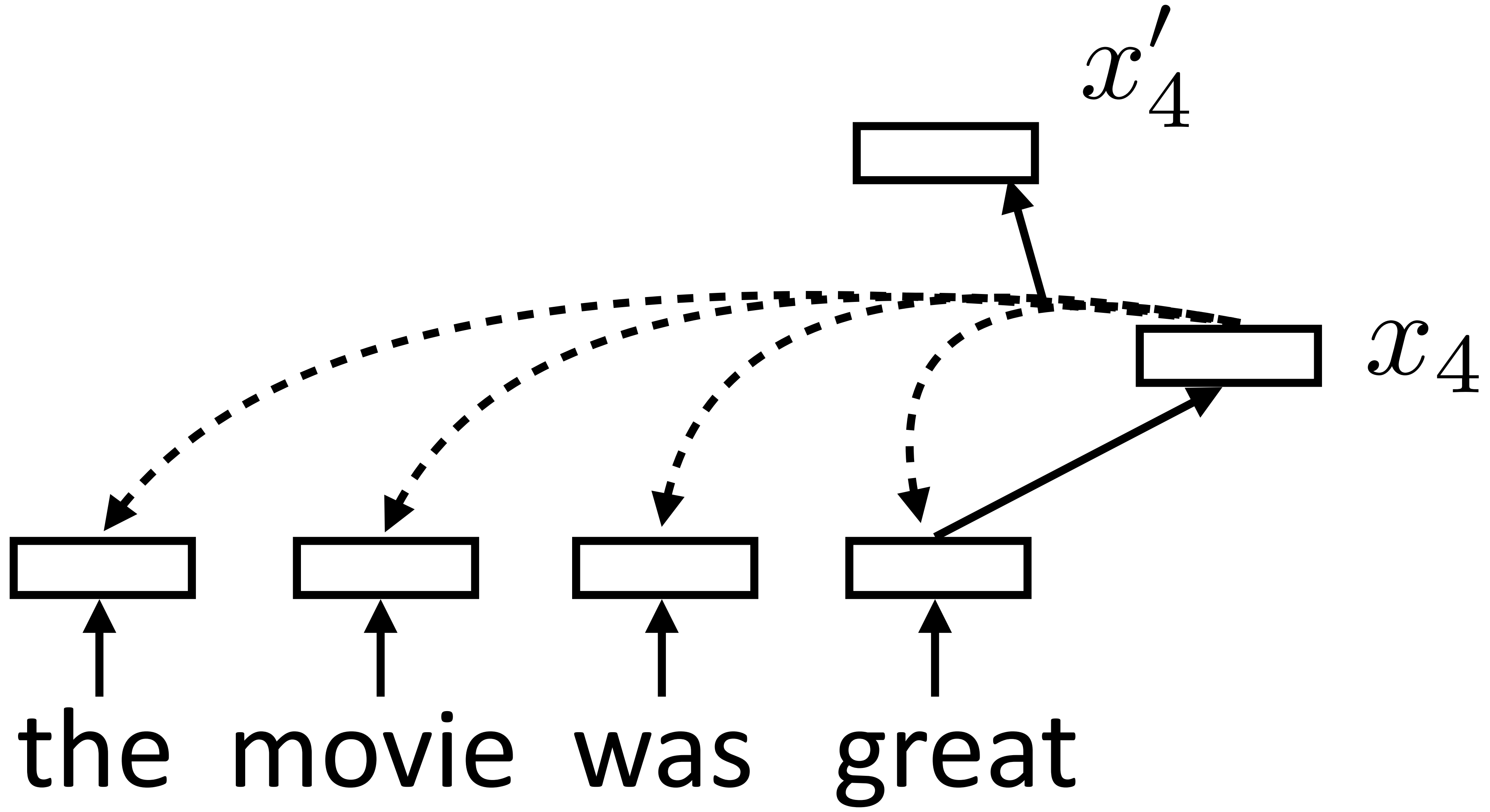
- Each word forms a "query" which then computes attention over each word



Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

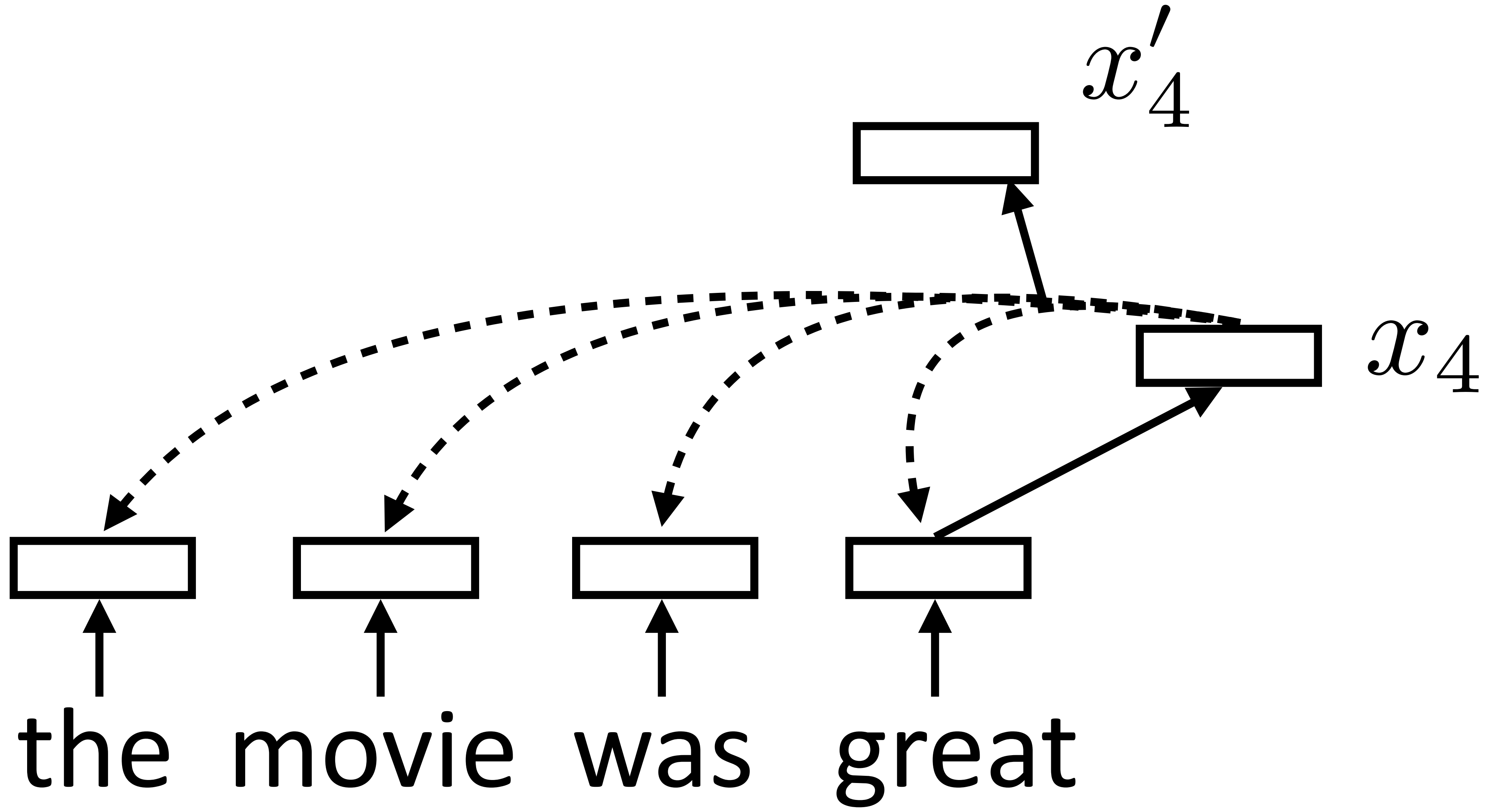


Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \text{softmax}(x_i^\top x_j) \quad \text{scalar}$$



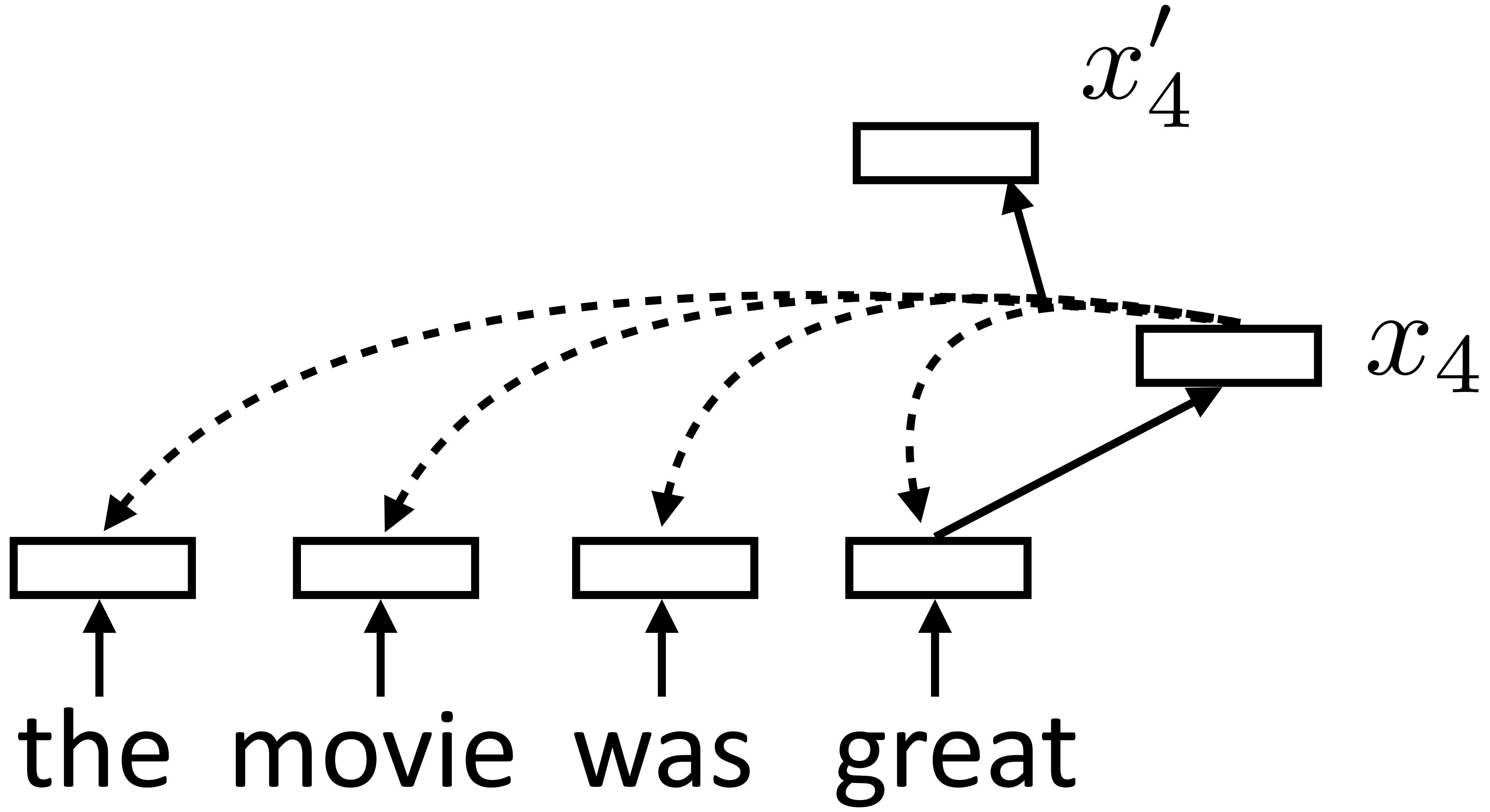
Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



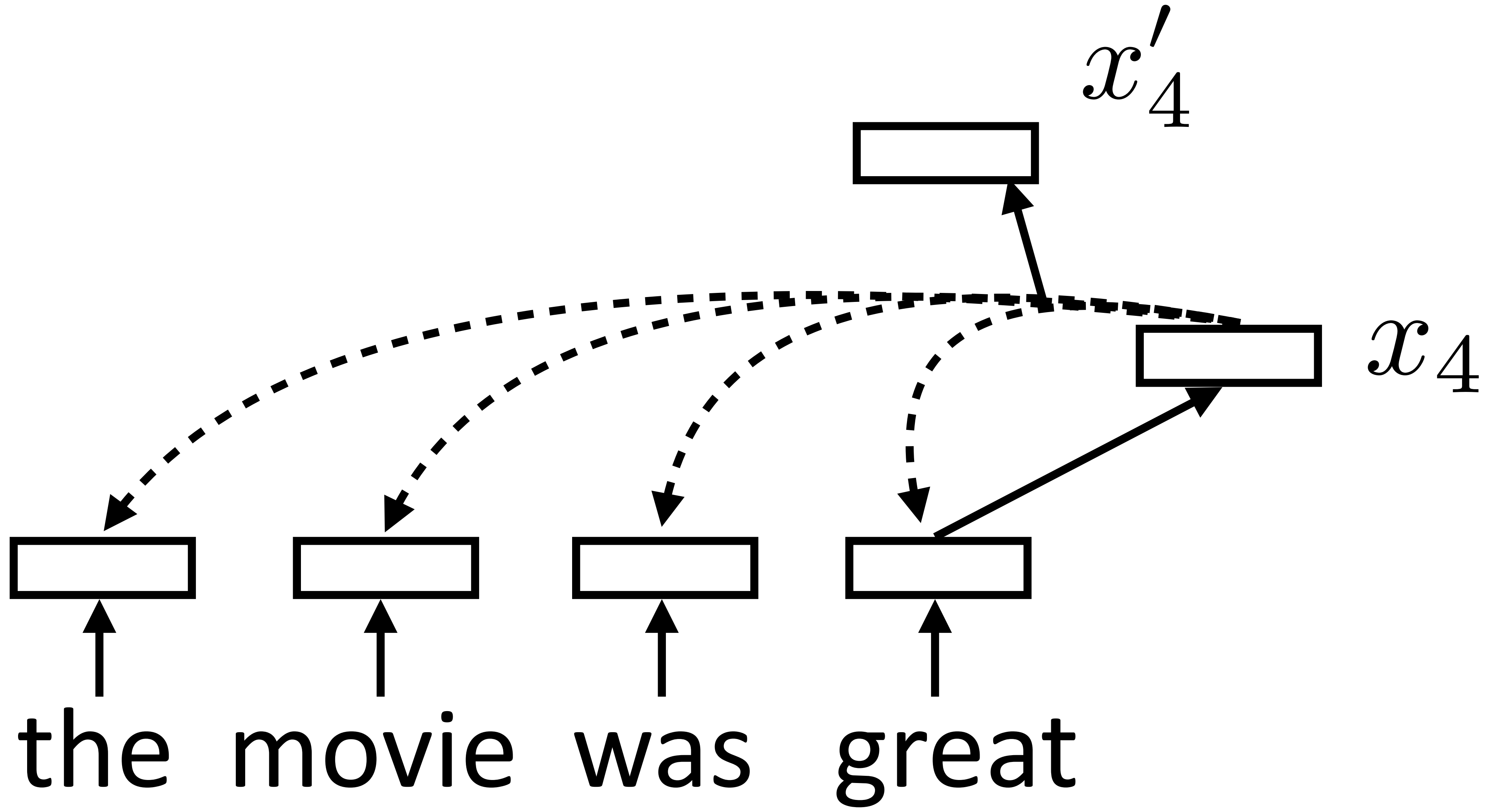
Vaswani et al. (2017)

Self-Attention

- Each word forms a “query” which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j) \quad \text{scalar}$$

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar * vector}$$



- Multiple “heads” analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

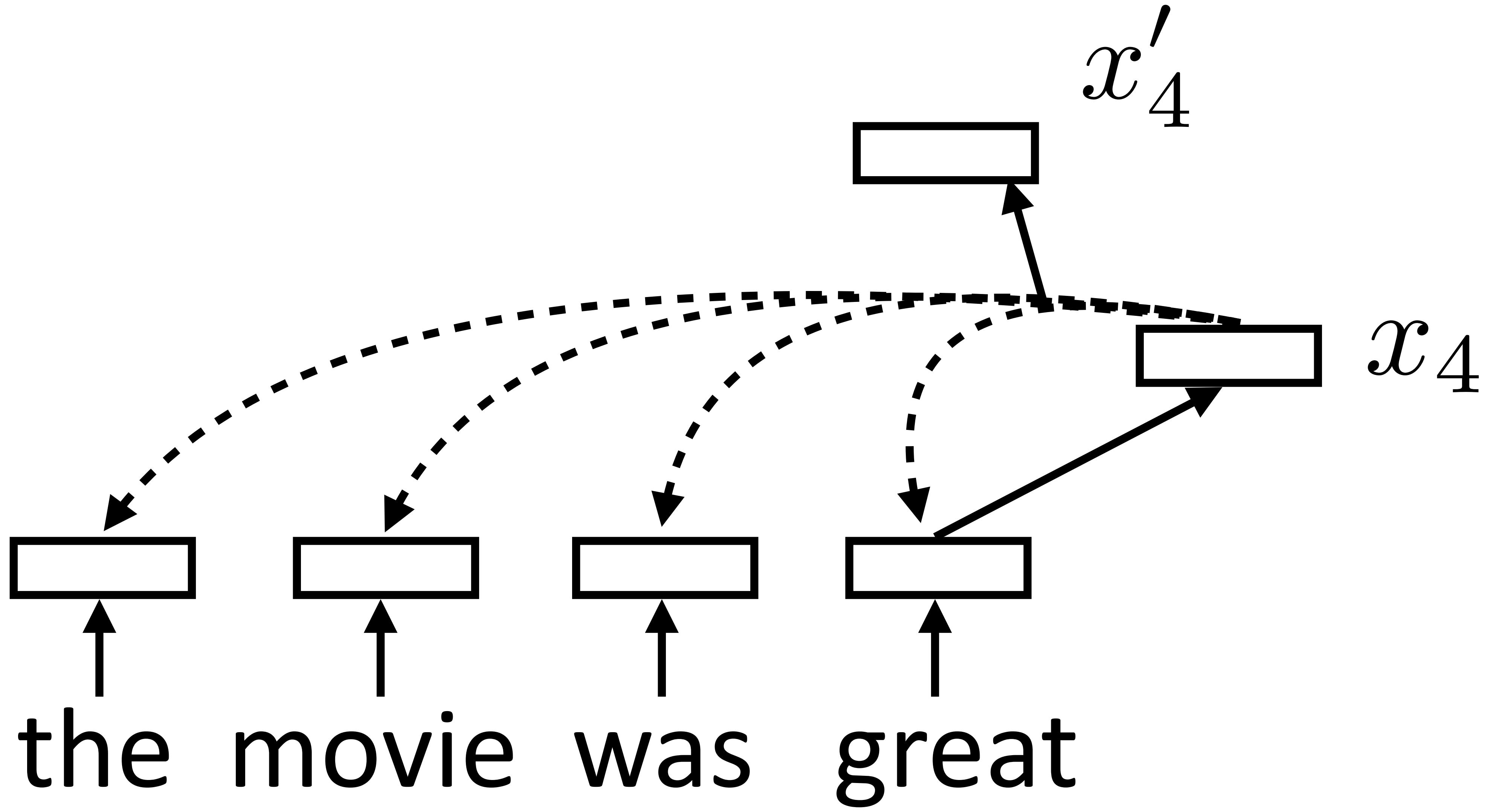
Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



- Multiple "heads" analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

$$\alpha_{k,i,j} = \mathrm{softmax}(x_i^\top W_k x_j)$$

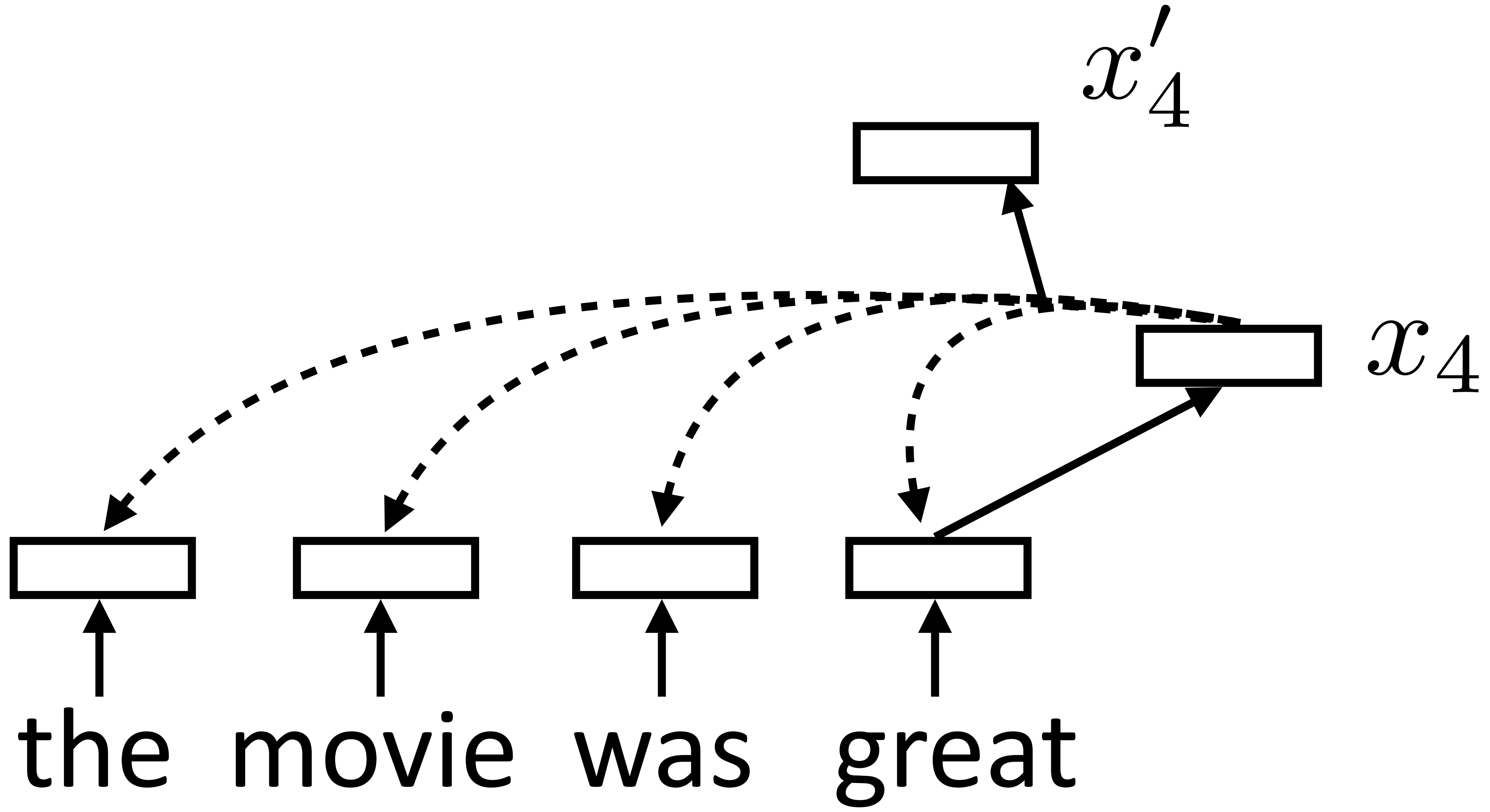
Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j) \quad \text{scalar}$$

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar * vector}$$



- Multiple "heads" analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

$$\alpha_{k,i,j} = \mathrm{softmax}(x_i^\top W_k x_j) \qquad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

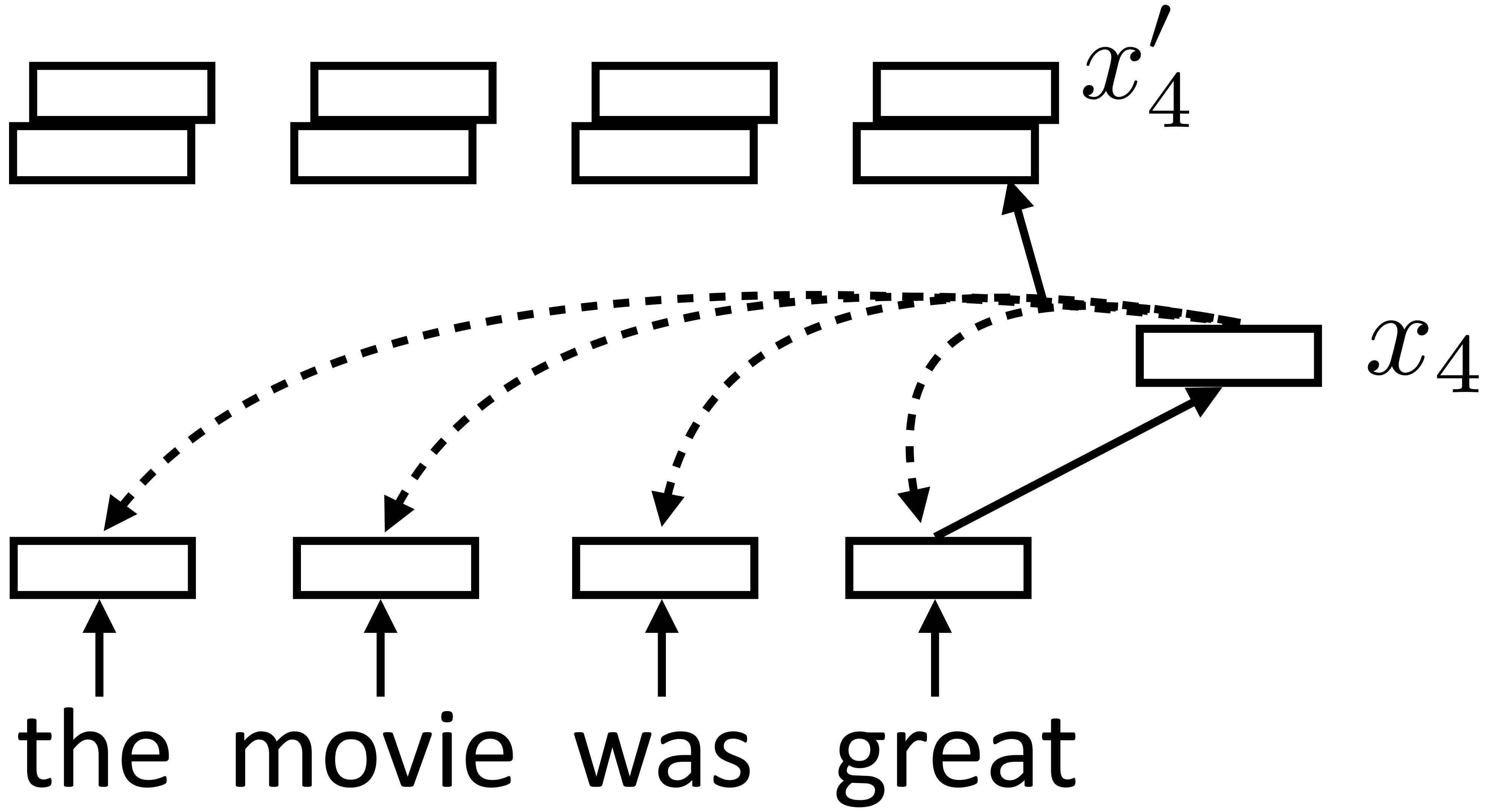
Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j) \quad \text{scalar}$$

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar * vector}$$

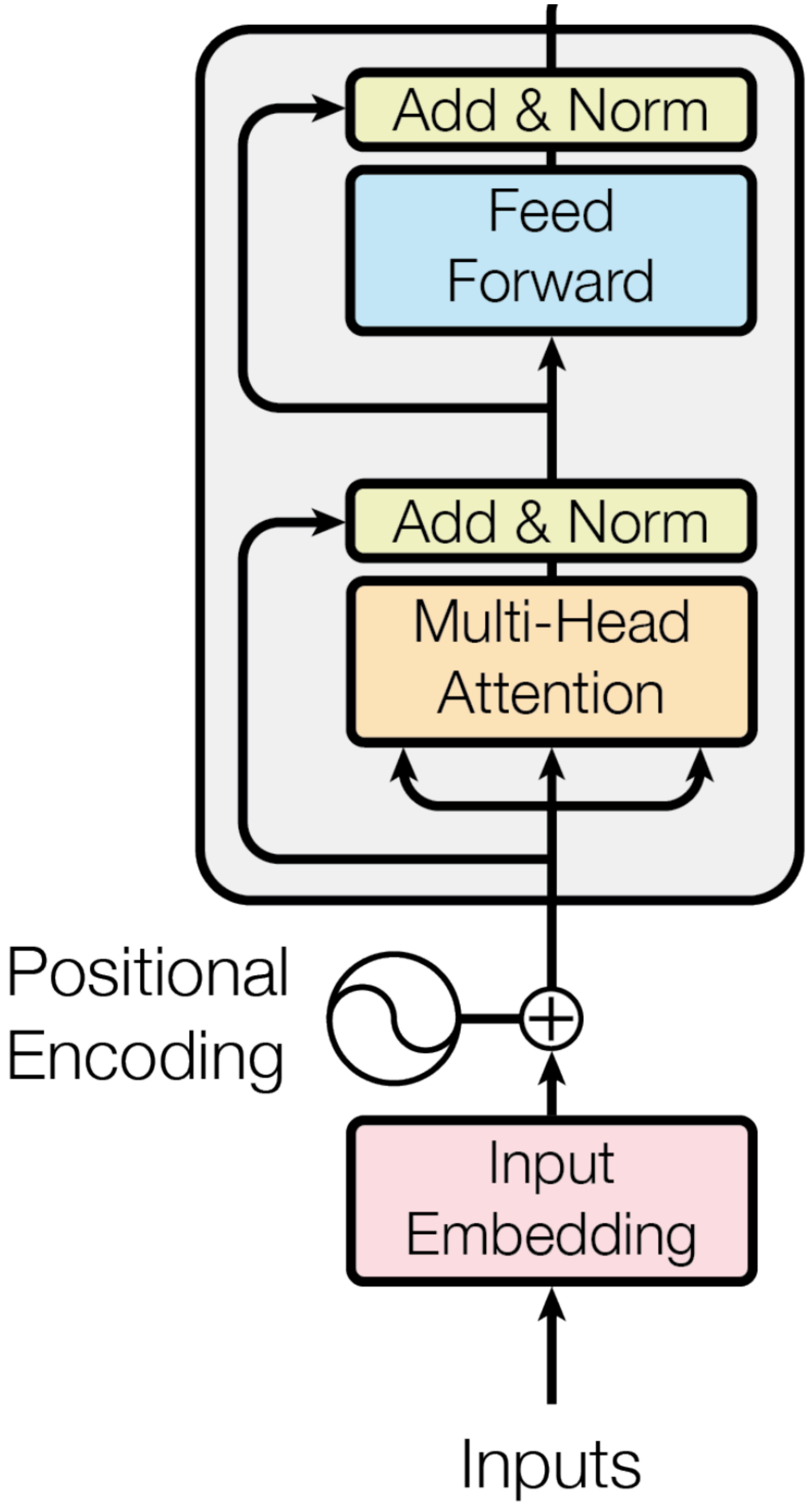


- Multiple "heads" analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

$$\alpha_{k,i,j} = \mathrm{softmax}(x_i^\top W_k x_j) \quad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

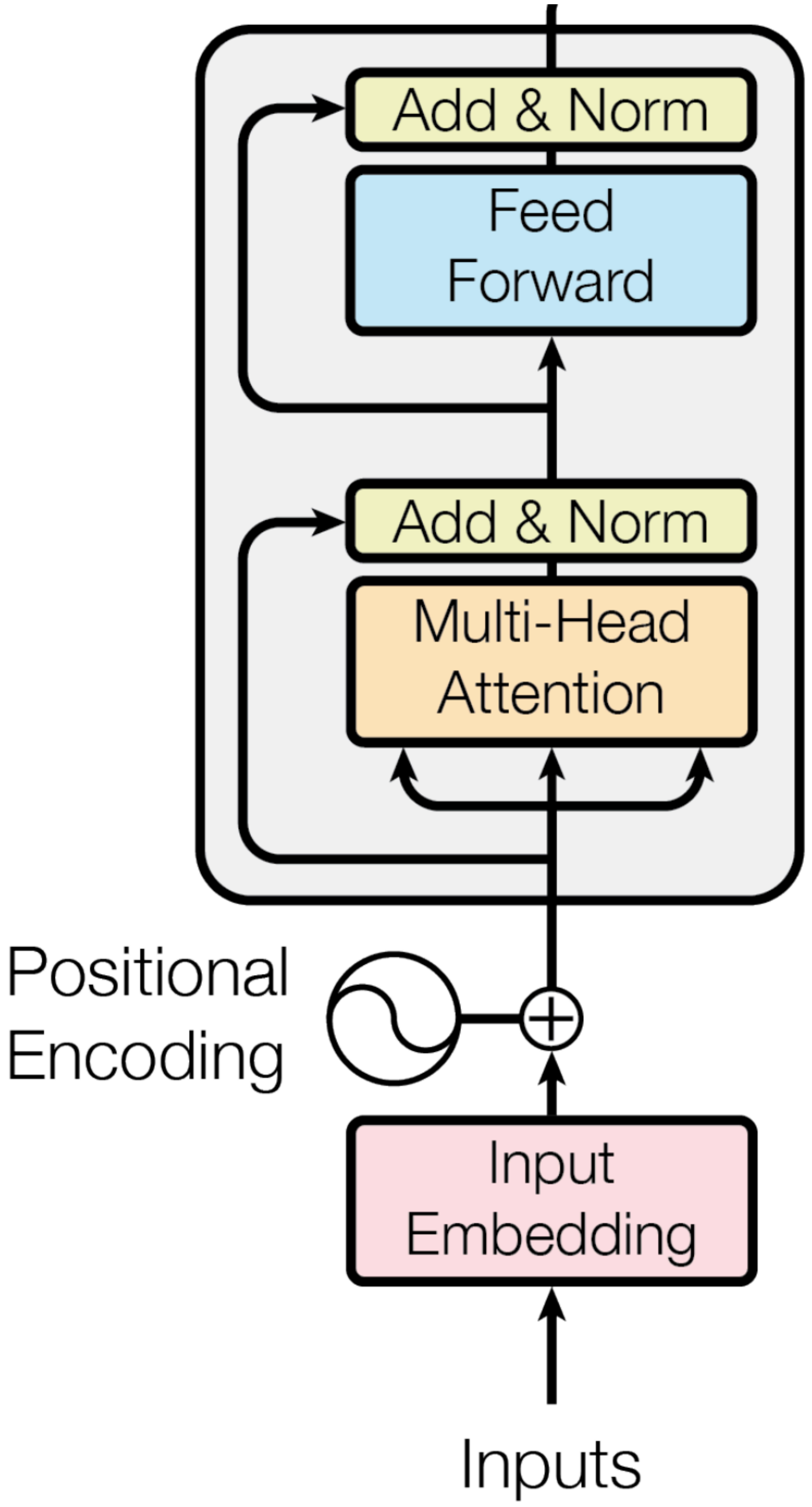
Vaswani et al. (2017)

Transformers



Vaswani et al. (2017)

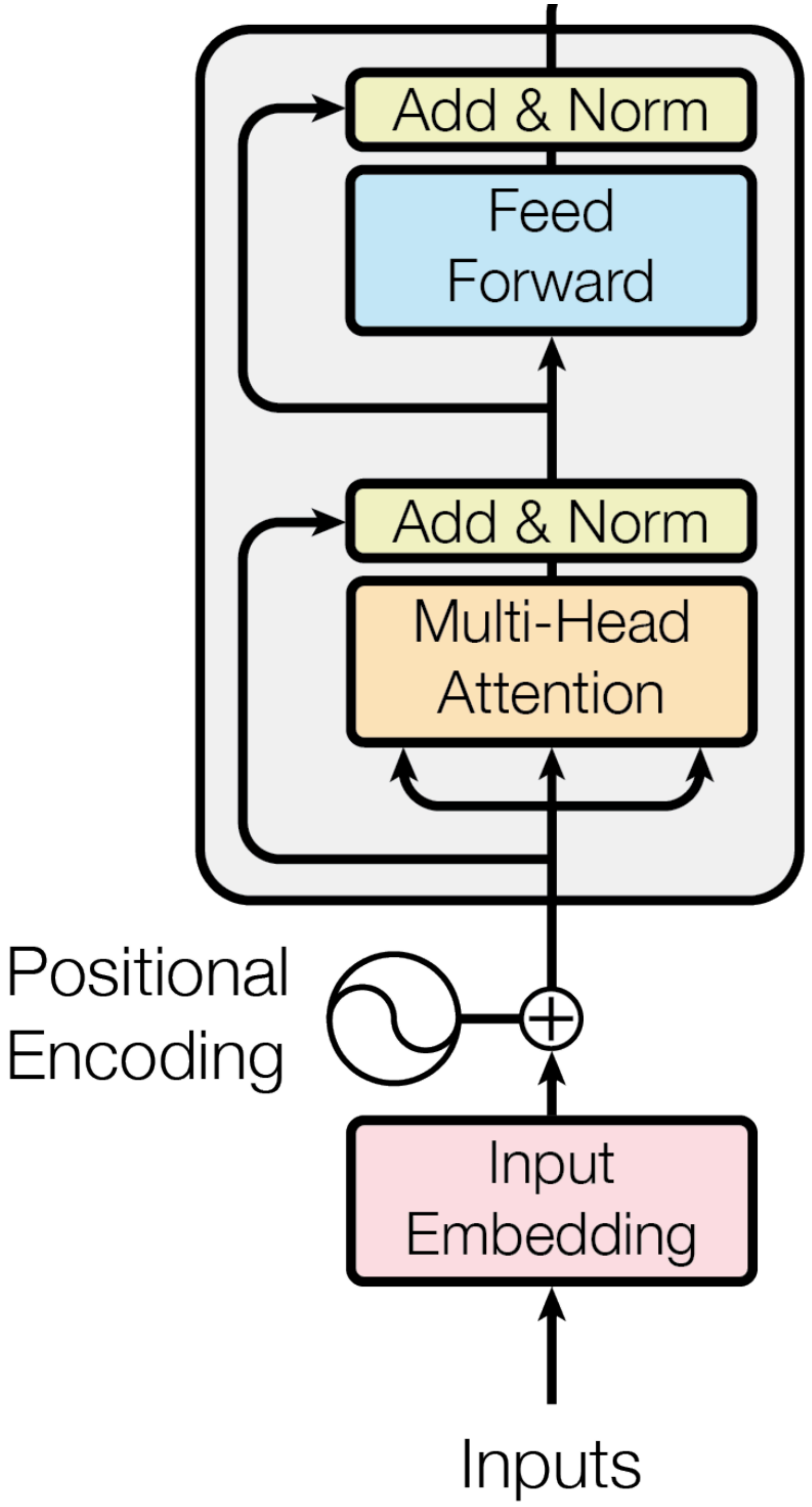
Transformers

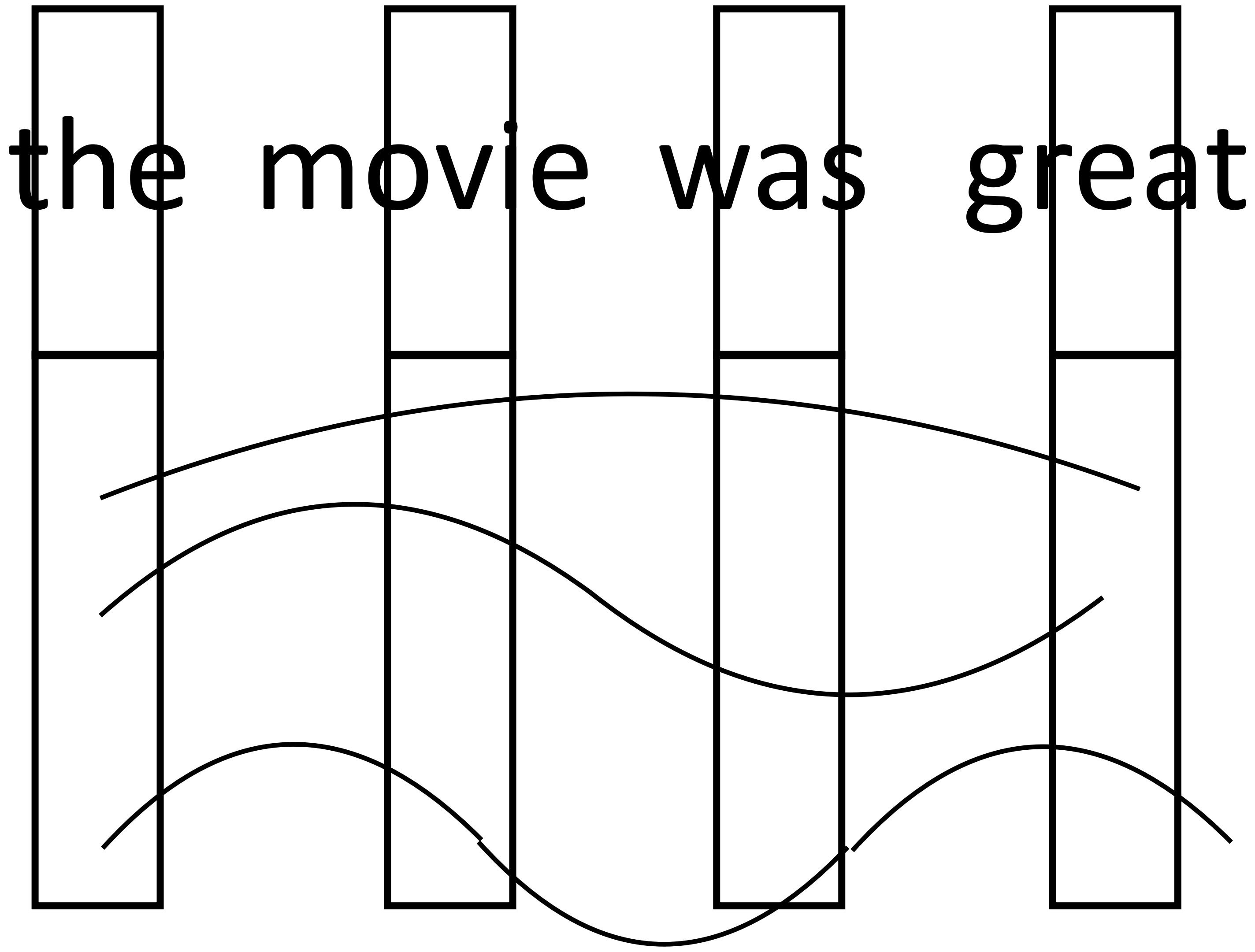


- Positional encoding: augment word embedding with position embeddings, each dim is a sine wave of a different frequency. Closer points = higher dot products

Vaswani et al. (2017)

Transformers

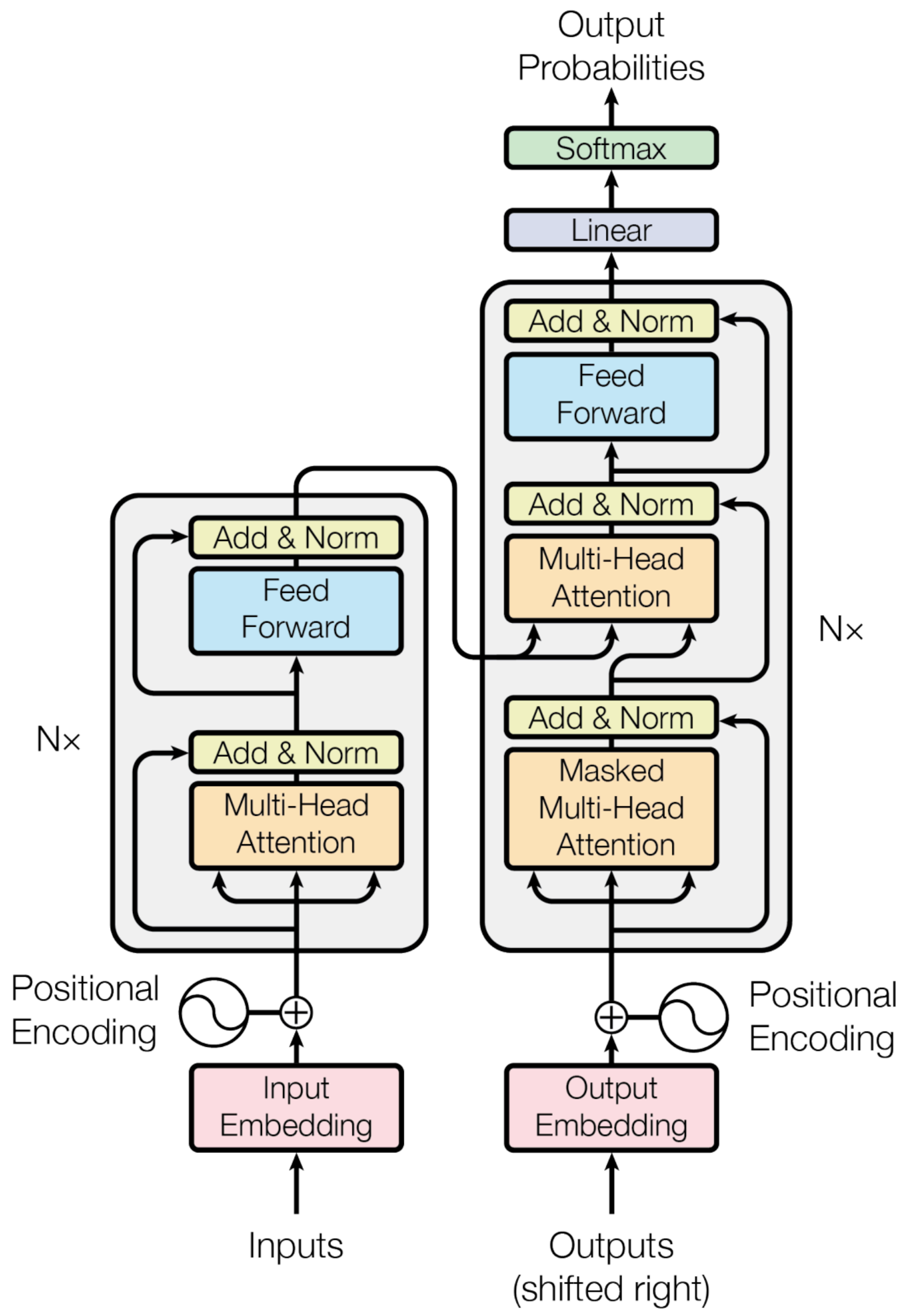




- Positional encoding: augment word embedding with position embeddings, each dim is a sine wave of a different frequency. Closer points = higher dot products

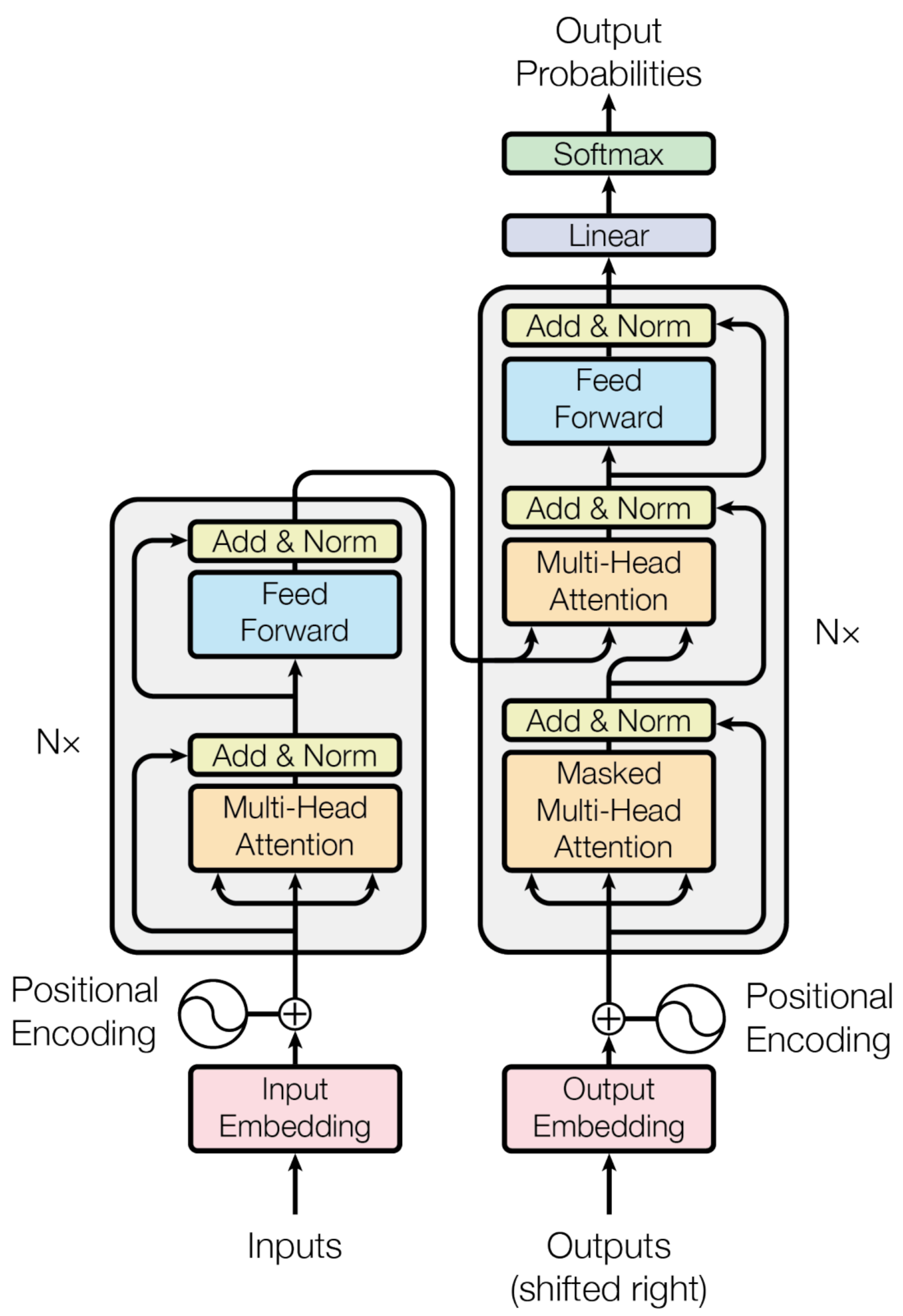
Vaswani et al. (2017)

Transformers



Vaswani et al. (2017)

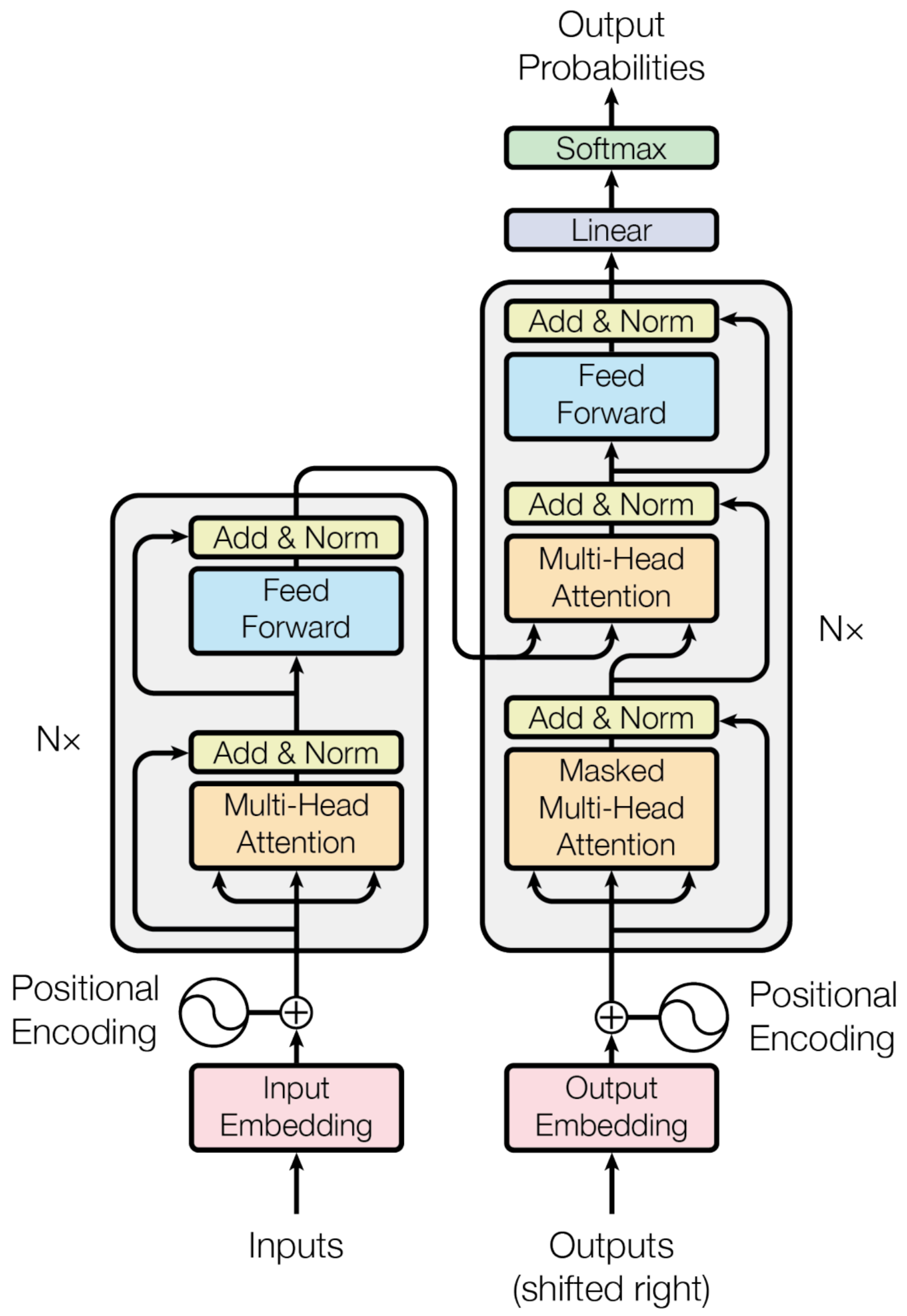
Transformers



- Encoder and decoder are both transformers

Vaswani et al. (2017)

Transformers



- Encoder and decoder are both transformers
- Decoder consumes the previous generated token (and attends to input), but has *no recurrent state*

Vaswani et al. (2017)

Transformers

Model	BLEU	
	EN-DE	EN-FR
ByteNet [18]	23.75	
Deep-Att + PosUnk [39]		39.2
GNMT + RL [38]	24.6	39.92
ConvS2S [9]	25.16	40.46
MoE [32]	26.03	40.56
Deep-Att + PosUnk Ensemble [39]		40.4
GNMT + RL Ensemble [38]	26.30	41.16
ConvS2S Ensemble [9]	26.36	**41.29**
Transformer (base model)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

Vaswani et al. (2017)

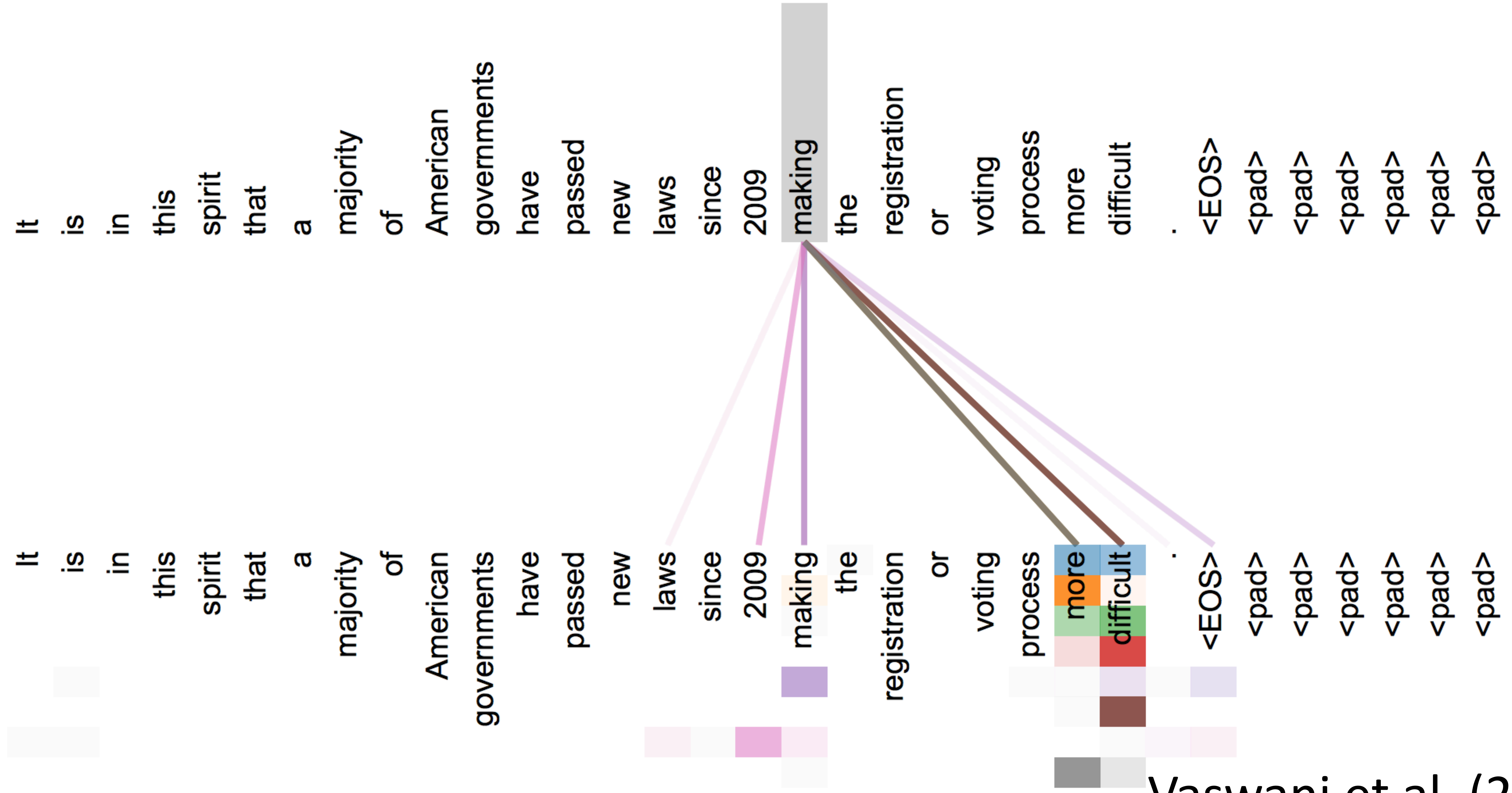
Transformers

Model	BLEU	
	EN-DE	EN-FR
ByteNet [18]	23.75	
Deep-Att + PosUnk [39]		39.2
GNMT + RL [38]	24.6	39.92
ConvS2S [9]	25.16	40.46
MoE [32]	26.03	40.56
Deep-Att + PosUnk Ensemble [39]		40.4
GNMT + RL Ensemble [38]	26.30	41.16
ConvS2S Ensemble [9]	26.36	**41.29**
Transformer (base model)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

- Big = 6 layers, 1000 dim for each token, 16 heads, base = 6 layers + other params halved

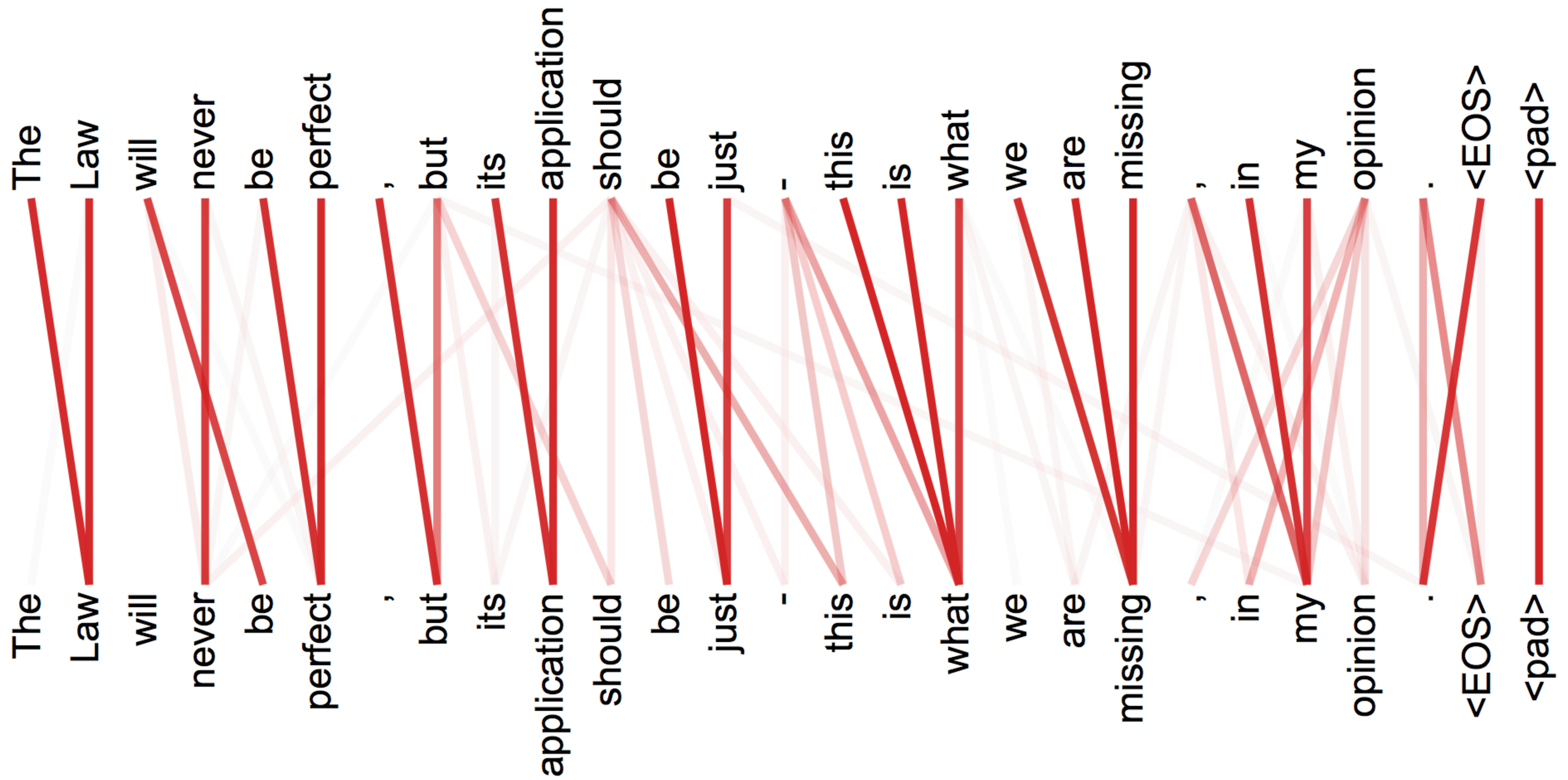
Vaswani et al. (2017)

Visualization



Vaswani et al. (2017)

Visualization



Vaswani et al. (2017)

Takeaways

- Can build MT systems with LSTM encoder-decoders, CNNs, or transformers
- Word piece / byte pair models are really effective and easy to use
- State of the art systems are getting pretty good, but lots of challenges remain, especially for low-resource settings