Neural Networks, Part 4

Philipp Koehn

23 April 2015

Big Picture

- Use of neural networks has led to significant improvements

- Incremental strategy:
  replace statistical components with neural components

- Leap forward strategy:
  start from scratch: neural machine translation

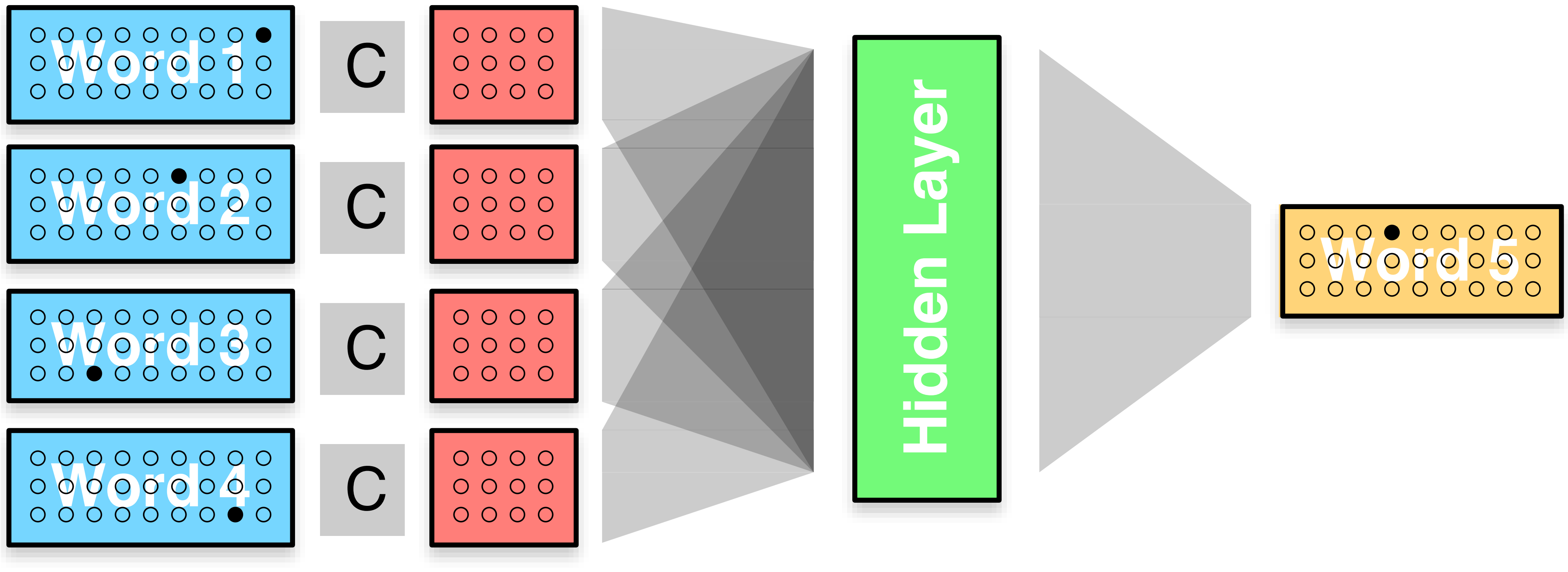
Neural Components

- Word alignment (Tamura et al., 2014)
- Language model
- Phrase translation
- Operation sequence model
- Reordering
- Morphological prediction (Tran et al., 2014)
- Syntactic models

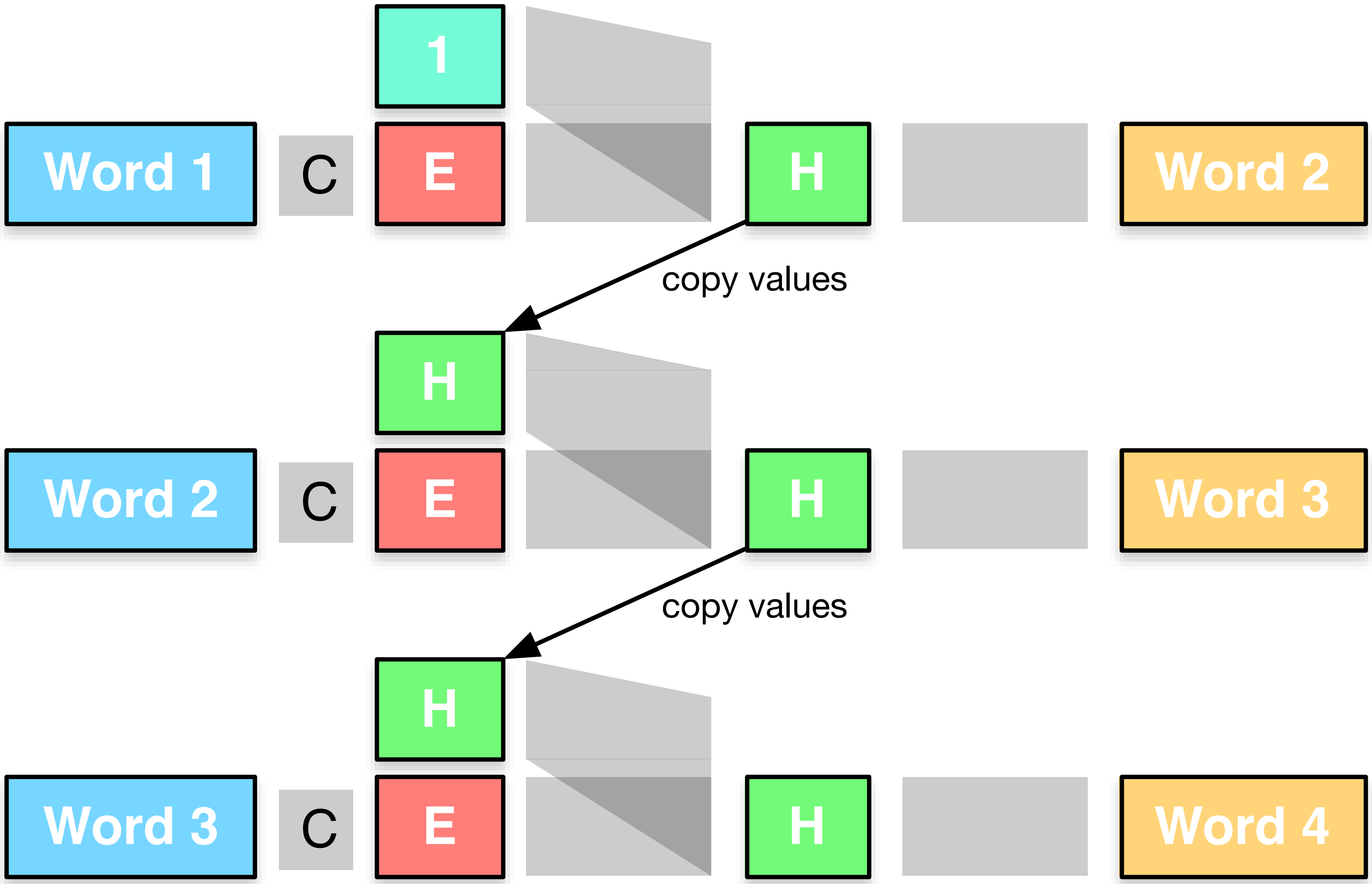
Language Models

- We discussed this last week
- Modeling variants
  - feed-forward neural network
  - recurrent neural network
  - long short term memory neural network
- May include source context

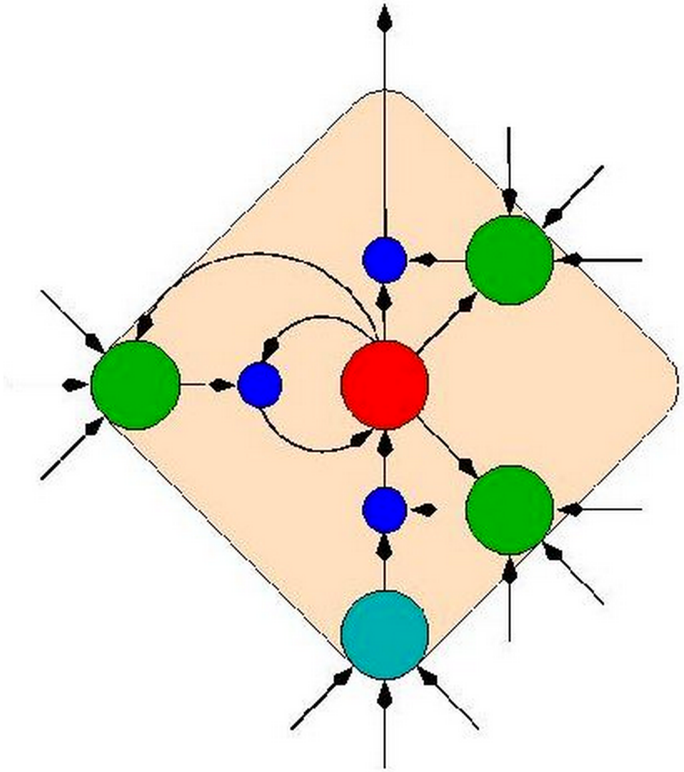
Feed Forward Neural Network



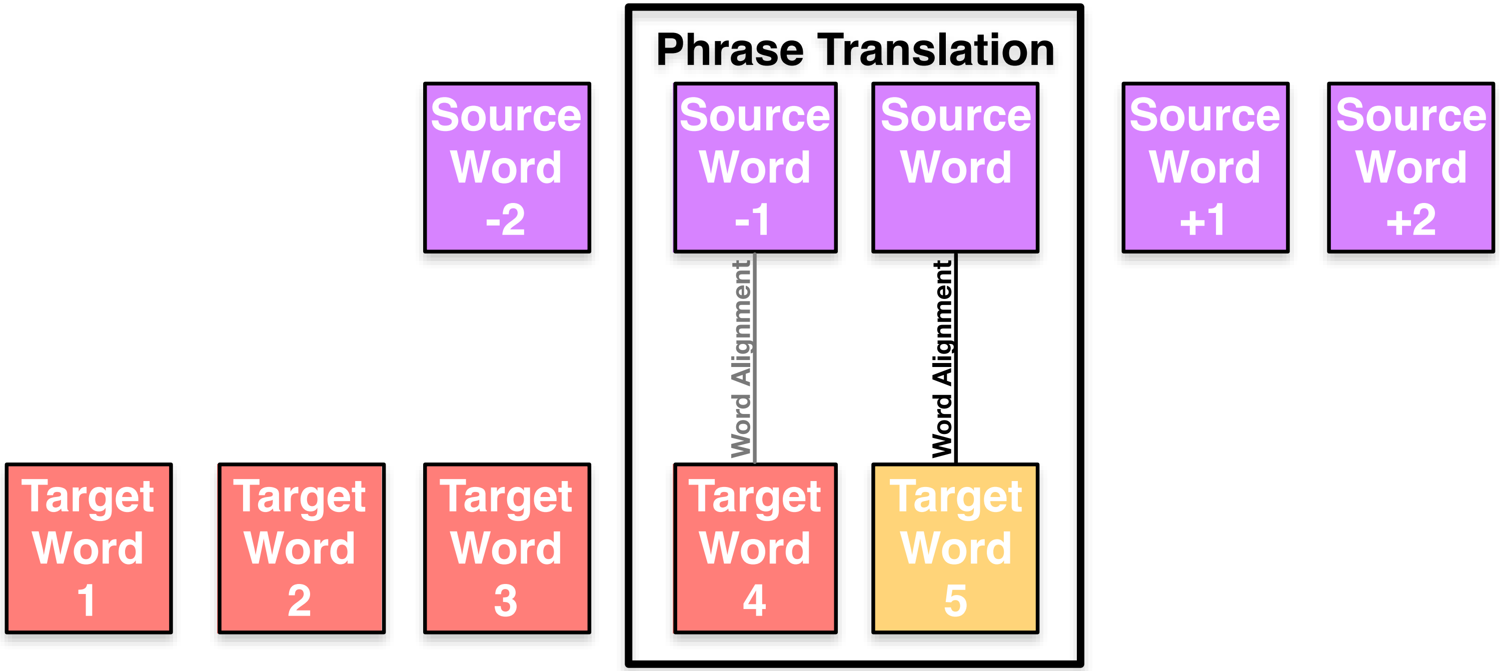
Recurrent Neural Network



Long Short Term Memory



Adding Source Context (Devlin et al., 2014)



- Normal 5-gram language model

$$p(e_5|e_1, e_2, e_3, e_4)$$

- 5-gram language model with source context

$$p(e_5|e_1, e_2, e_3, e_4, f_{a(5)-2}, f_{a(5)-1}, f_{a(5)}, f_{a(5)+1}, f_{a(5)+2})$$

phrase translation

Phrase Translation

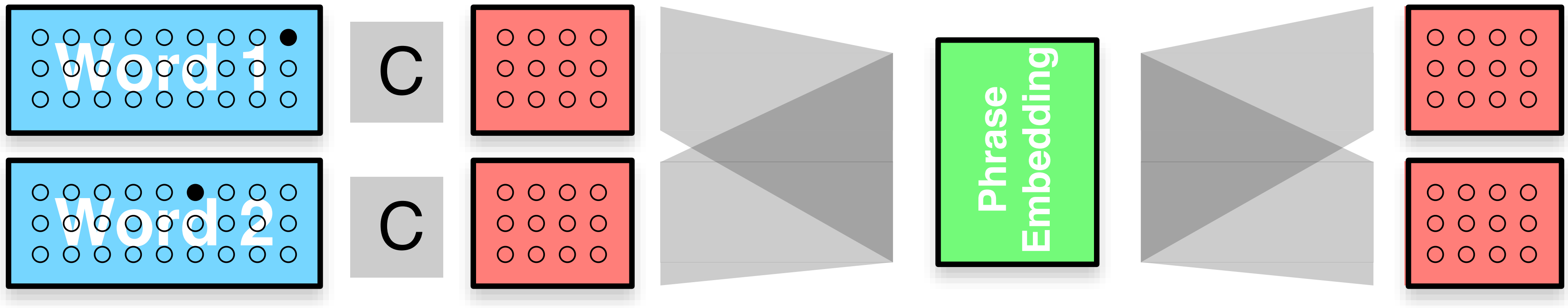
- Atomic unit of translation: phrase mapping
  - große Haus → big house
  - eine Tasse → a cup of
- Probability distribution

$$\phi(\bar{e}|\bar{f}) = \frac{\text{count}(\bar{e},\bar{f})}{\text{count}(\bar{f})}$$

- Smoothed with lexical translation probabilities
- Convert this into a neural network

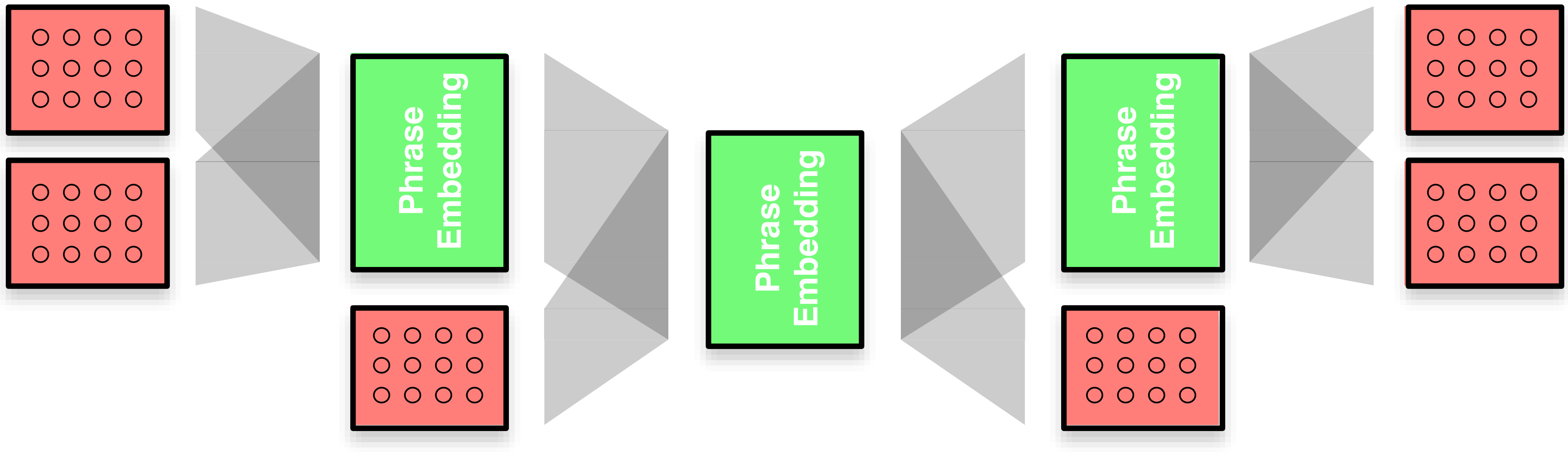
How to Encode a Bigram

- Auto-encoder (for bigram)



- Obtain word embeddings by traditional means (NNLM)
- Map embeddings of 2 words into lower-dimensional space
  $\rightarrow$ phrase embedding
- Learn to reconstruct the words

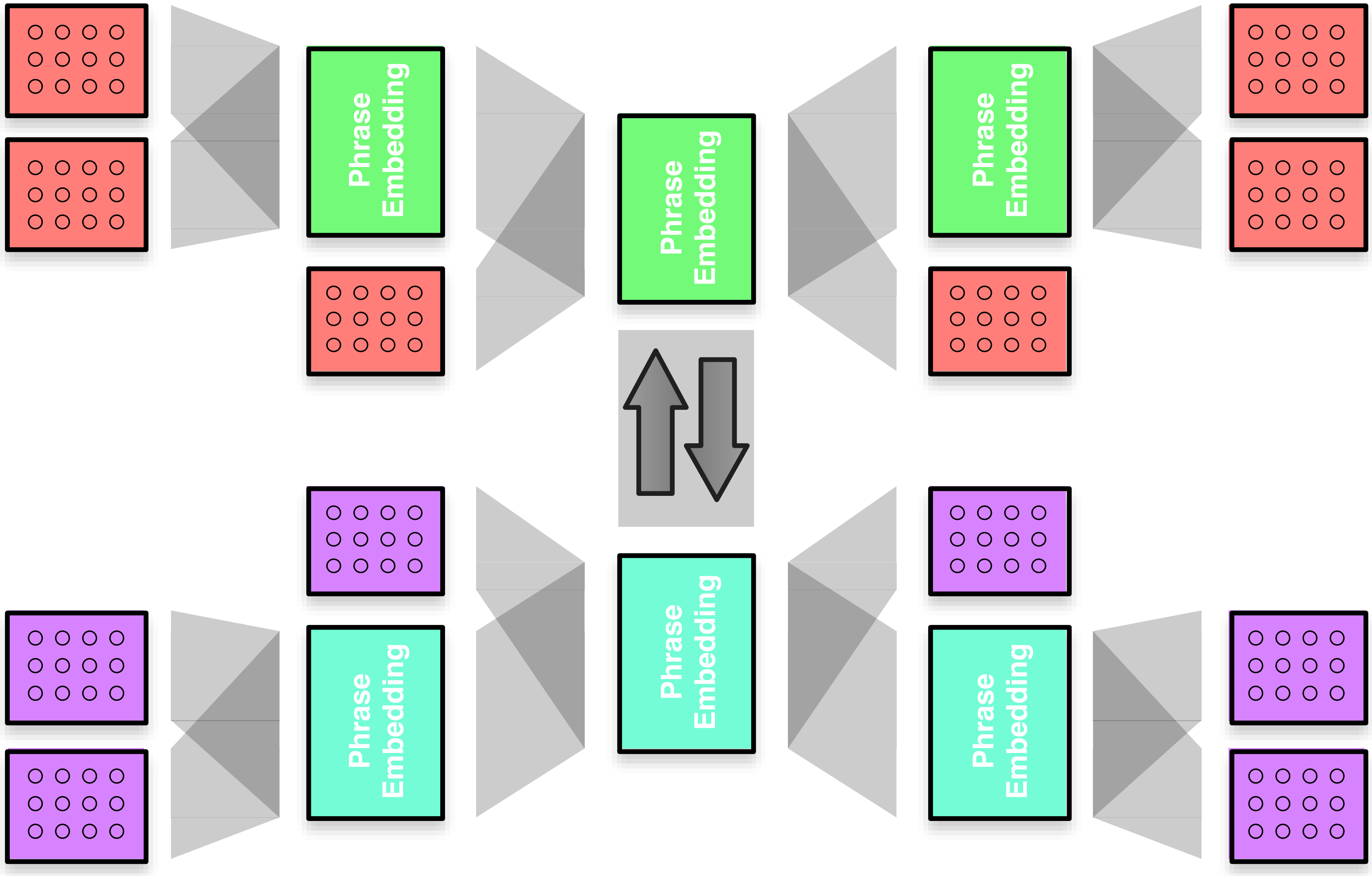
Recursive Auto-Encoder



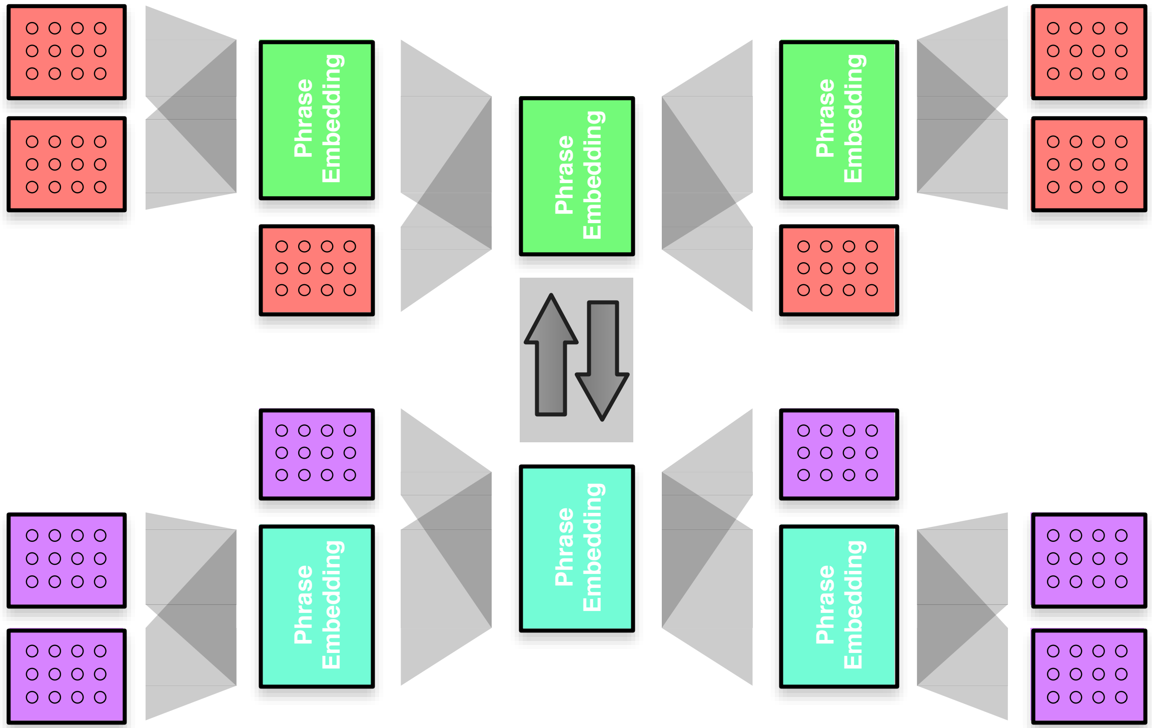
- Recursive: combine phrase embedding and word

- Same weights for
  - word+word → phrase → word+word
  - phrase+word → phrase → phrase+word

Phrase Translation

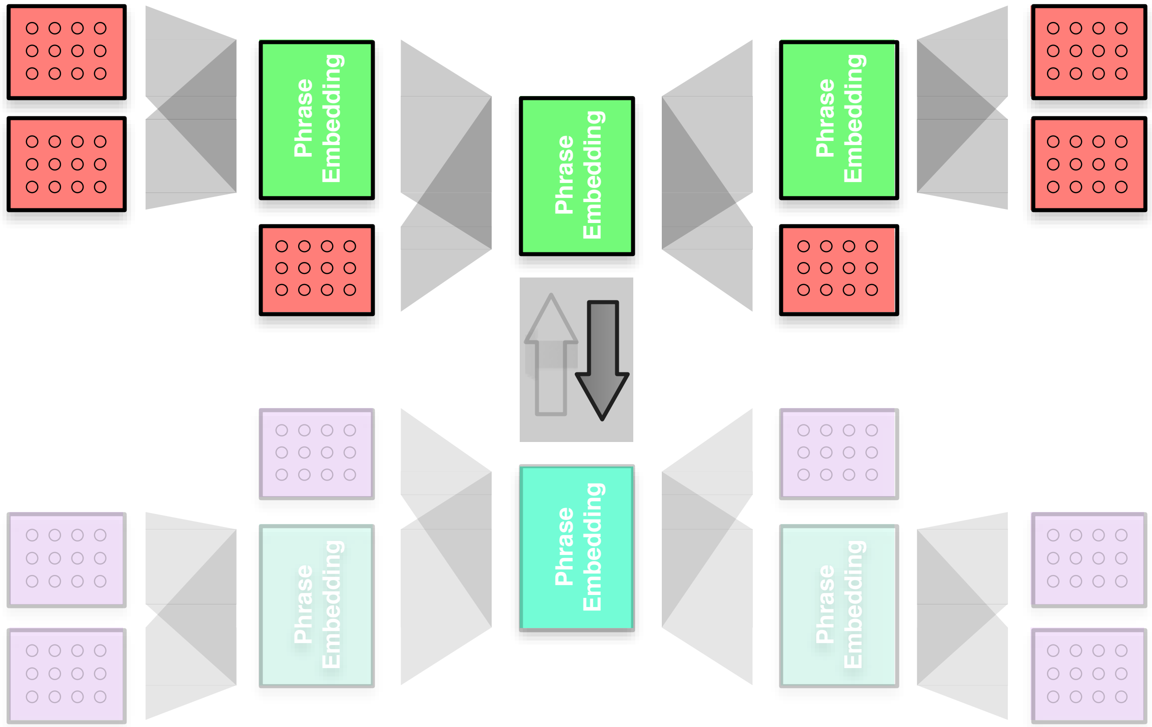


Training



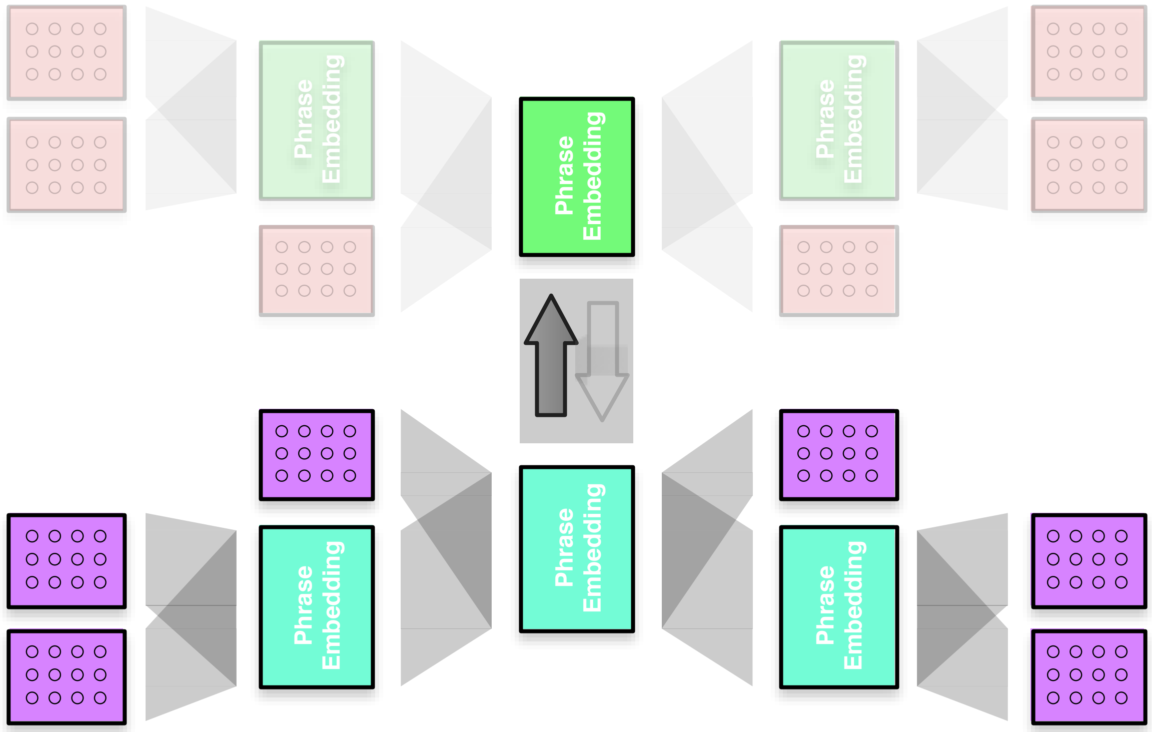
- 2 optimization objectives
  - reconstruction error in auto encoder
  - phrase translation error

Training



- Alternate between
  - **training source embedding / translation to target**
  - training target embedding / translation to source

Training



- Alternate between
  - training source embedding / translation to target
  - **training target embedding / translation to source**

Integration into Decoder

- Strictly tied to existing phrase table
- No use of additional context

$\Rightarrow$ Use as an additional feature function

$\Rightarrow$ Use to filter out bad phrase pairs

operation sequence model

Operation Sequence Model

o_1	Generate(natürlich, of course)	natürlich ↓ of course
o_2 o_3	Insert Gap Generate (John, John)	natürlich ↓ $\boxed{}$ John of course John
o_4 o_5	Jump Back (1) Generate (hat, has)	natürlich hat ↓ John of course John has
o_6	Jump Forward	natürlich hat John ↓ of course John has
o_7	Generate(natürlich, of course)	natürlich hat John Spaß ↓ of course John has fun
o_8 o_9	Generate(am, with) GenerateTargetOnly(the)	natürlich hat John Spaß am ↓ of course John has fun with the
o_{10}	Generate(Spiel, game)	natürlich hat John Spaß am Spiel ↓ of course John has fun with the game

Operation Sequence Model

- Phrase based models have problems with
  - phrase segmentation
  - balance of short and long phrases

- Break up phrase translation
  - minimal translation units
  - reordering operations

- Model a sequence of operations

$$p(o_1)\ p(o_2|o_1)\ p(o_3|o_1, o_2)\ ...\ p(o_{10}|o_6, o_7, o_8, o_9)$$

Neural Operation Sequence Model

- Not done yet

  almost: Hu et al. (2014) and Wu et al. (2014) model MTU sequences (recurrent neural network, only re-ranking)

- Arguably, OSM and Devlin et al. (2014)'s JNNLM do something similar

  as Birch et al. (2014) show:

	English–French	German–English
Baseline	35.7	32.5
OSM	37.3 (+1.6)	33.0 (+0.5)
JNNLM	36.7 (+1.0)	32.4 (–0.1)
OSM + JNNLM	37.4 (+1.7)	32.8 (+0.3)

reordering

Reordering (Li et al., 2014)

- Lexicalized reordering model

$$p(\text{orientation}|\bar{f}, \bar{e})$$

with orientation $\in$ {monotone, swap, discontinuous}

- Encode phrases with recursive auto-encoders

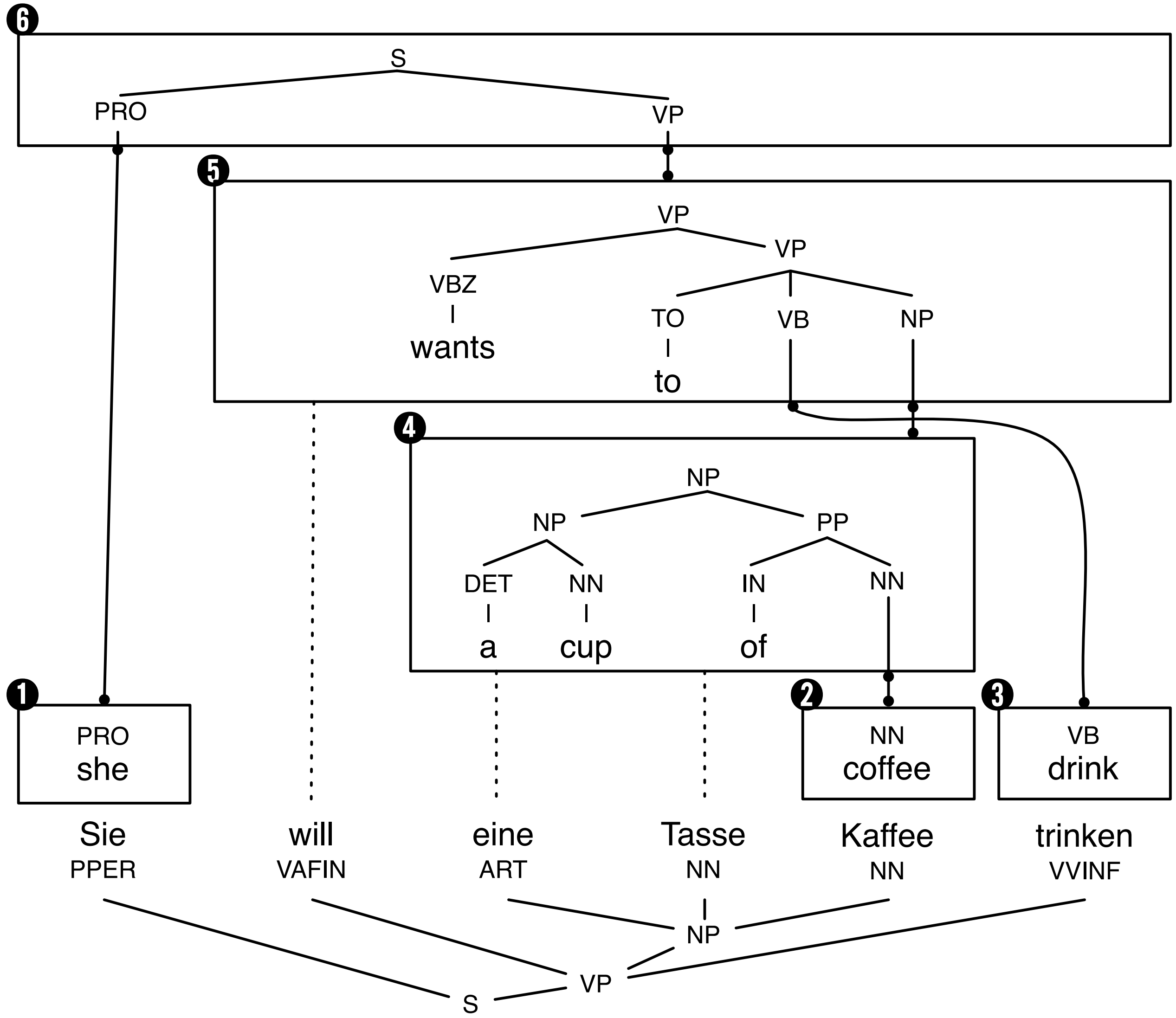
- We may also want to include the previous phrase pair

$$p(\text{orientation}|\bar{f}, \bar{e}, \bar{f}_{-1}, \bar{e}_{-1})$$

Richer context $\rightarrow$ only used for re-ranking

syntax models

Syntax Models



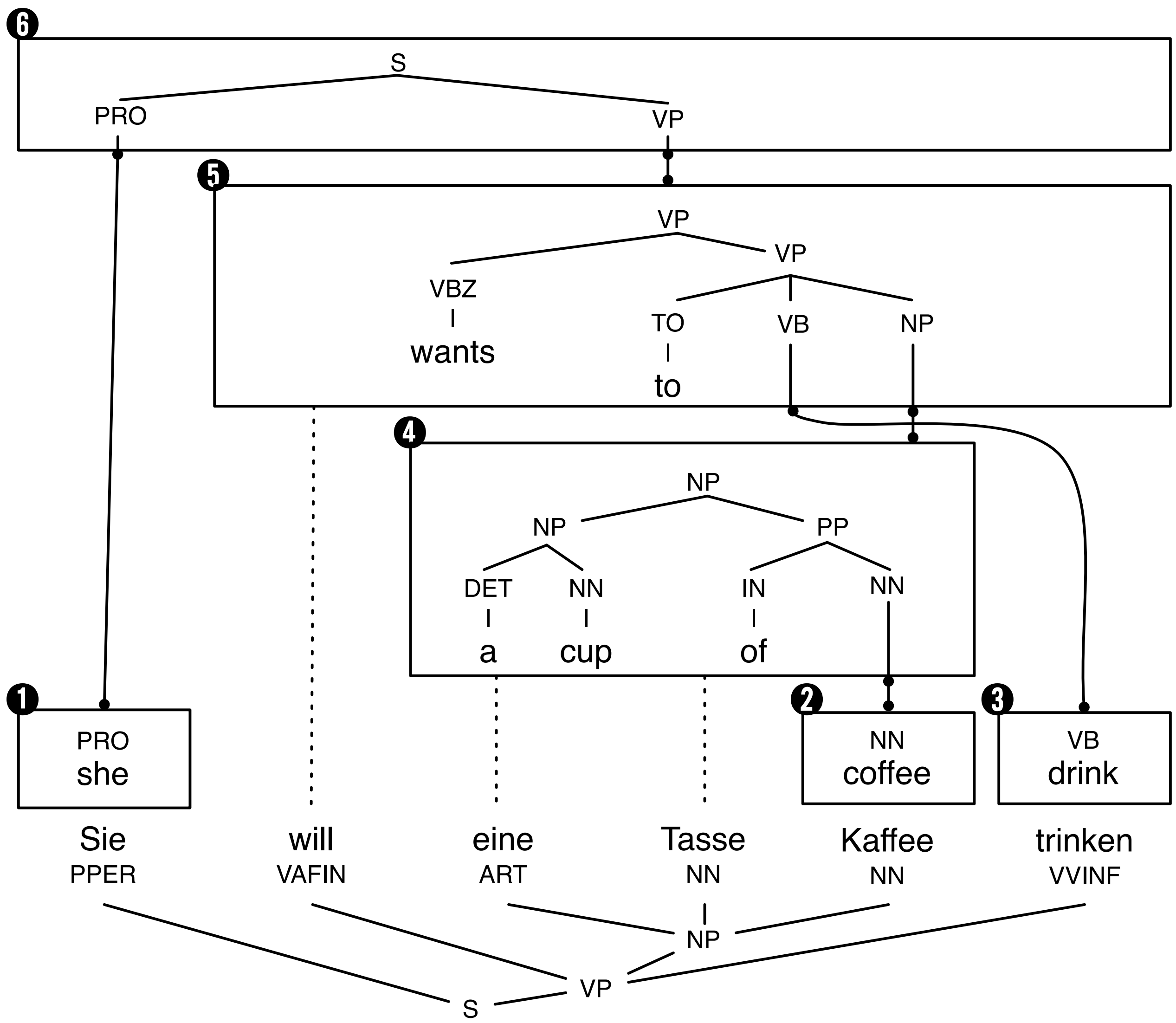
Directions

- Better transfer rules — not done yet:
  - better back off between minimal rules and composed rules
  - flexible use of source and target side syntax
  - long distance agreement

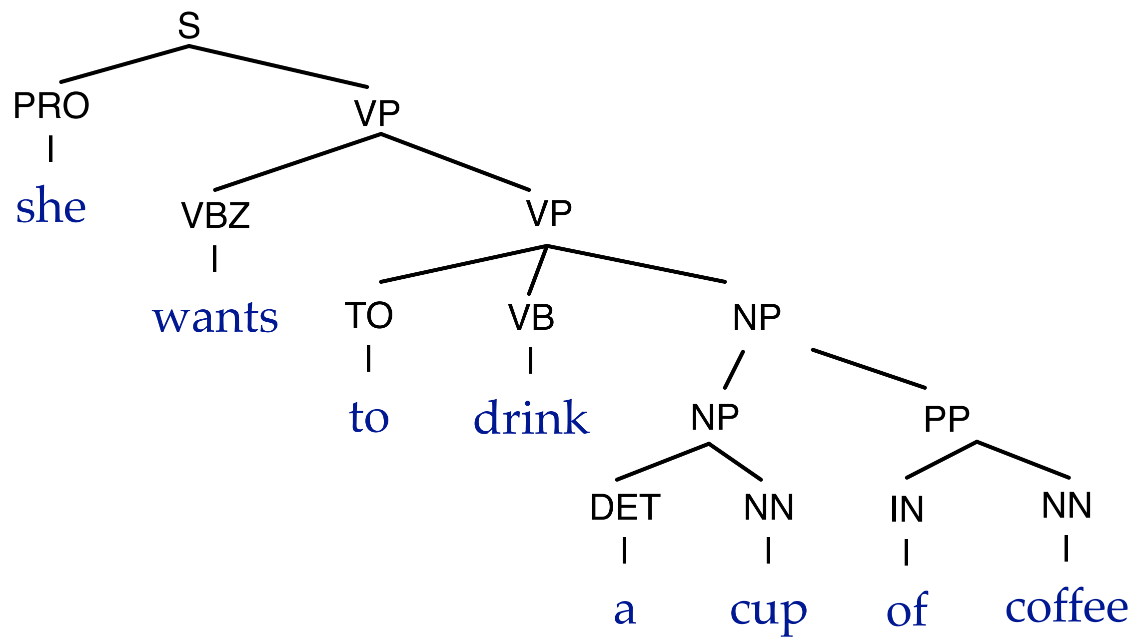
- Better syntactic language models
  - is the output syntactically coherent?

  $\rightarrow$ model the tree structure

Derivation Tree

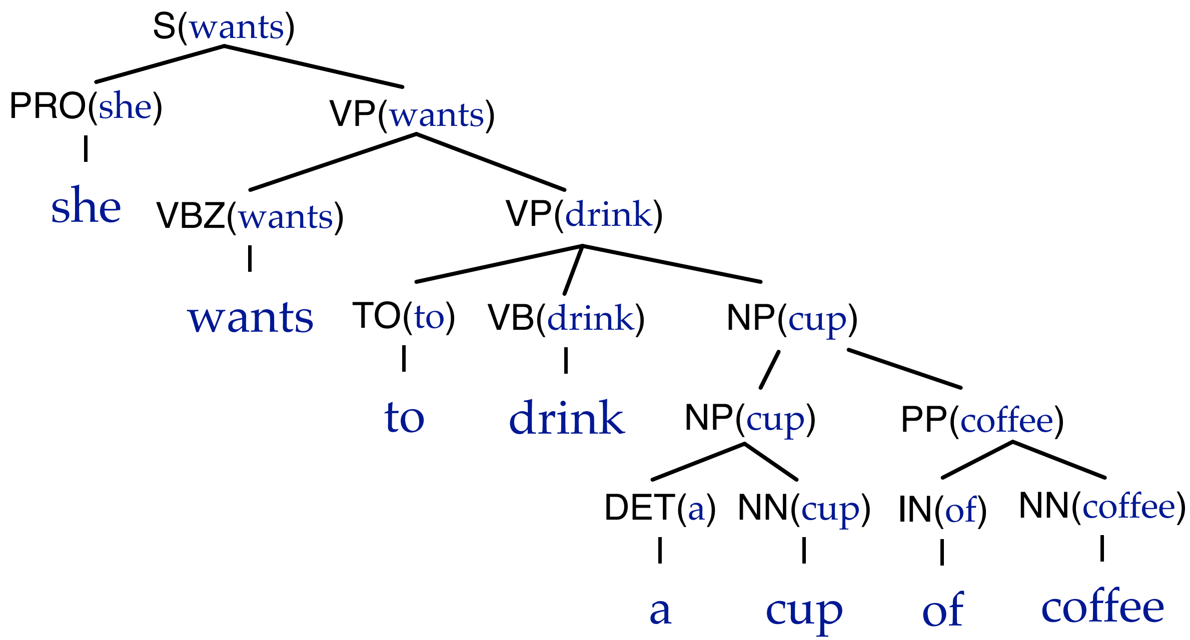


Phrase Structure Tree



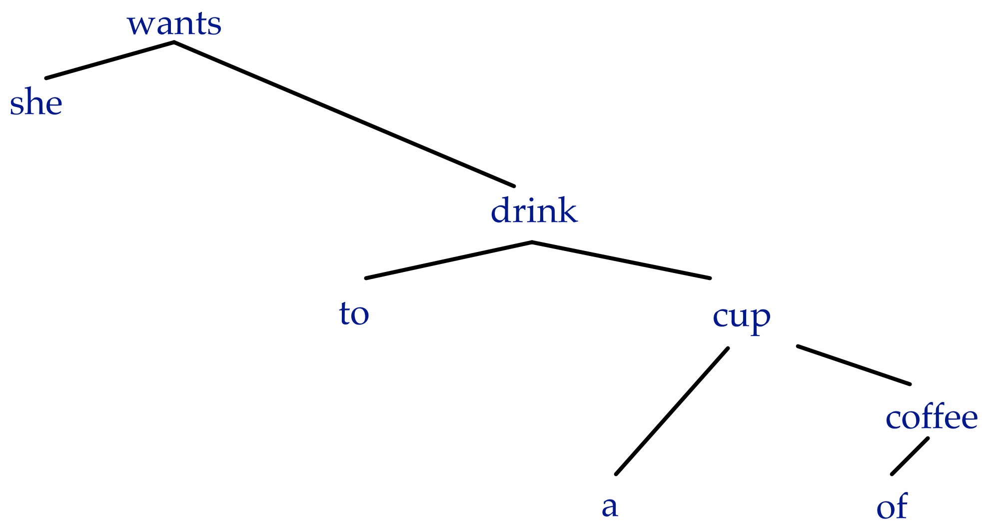
- Consider the phrase structure tree that was built

Head Words



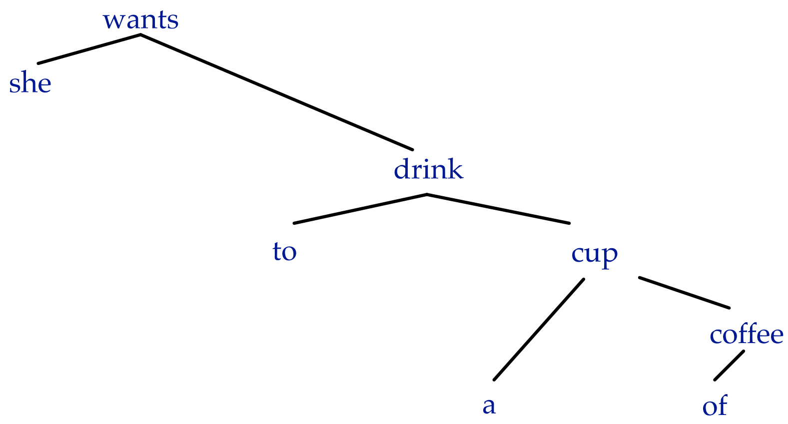
- Annotate with head words
  - standard rules which of the children is head node
  - e.g., noun phrase: last noun

Dependency Structure



- Reduce tree to non-inheriting children connections
- Parent / Grandparent relationships
  coffee → cup → drink
- Sibling relationships
  she ↔ drink

Dependency Model (Sennrich, 2015)



- Top-down / left-right model

- Predict from ancestry (up to 2)
  - parent
  - grand-parent

- Predict from left children (up to 2)

- Example: $p(\text{coffee}|\text{cup}, \text{drink}, \text{a}, \epsilon)$

Neural Network Model

- Probability distribution

$$p(\text{word}|\text{parent}, \text{grand-parent}, \text{left-most-sibling}, \text{2nd-left-most-sibling})$$

  for instance

$$p(\text{coffee}|\text{cup}, \text{drink}, \text{a}, \epsilon)$$

  can be converted straightforward into a feed-forward neural network

- Words encoded with embeddings

- Empty slots modeled by average embedding over all words

neural translation models

Encoder–Decoder Model

- Word embeddings seen as "semantic representations"
- Recurrent Neural Network
  $\rightarrow$ semantic representation of whole sentence
- Idea
  - encode semantics of the source sentence with recurrent neural network
  - decode semantics into target sentence from recurrent neural network
- Model

$$(w_1, ..., w_{l_f+l_e}) = (f_1, ..., f_{l_f}, e_1, ..., e_{l_e})$$

$$\prod_k p(w_1, ..., w_{l_f+l_e}) = \prod p(w_k | w_1, ..., w_{k-1})$$

- But: bias towards end of sentence

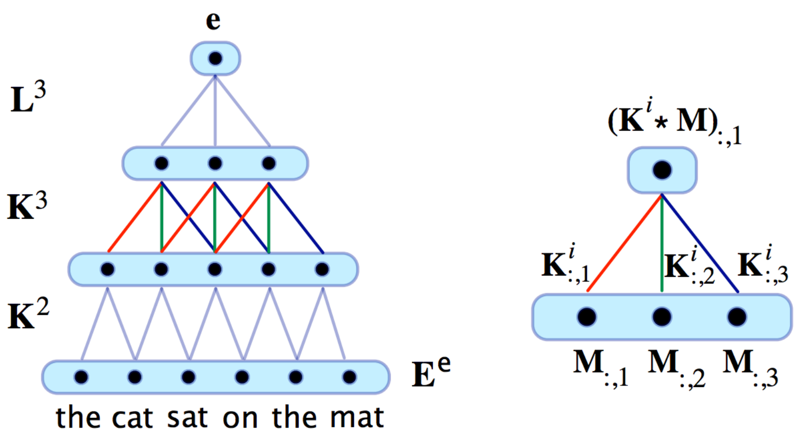
LSTM and Reversed Order (Sutskever et al., 2014)

- Long short term memory for better retention of long distance memory
- Reverse production of target sentence

$$(f_1, ..., f_{l_f}, e_{l_e}, ..., e_1)$$

- Some tricks (ensemble learning)
- Claims that it works as stand-alone model
  but better in reranking

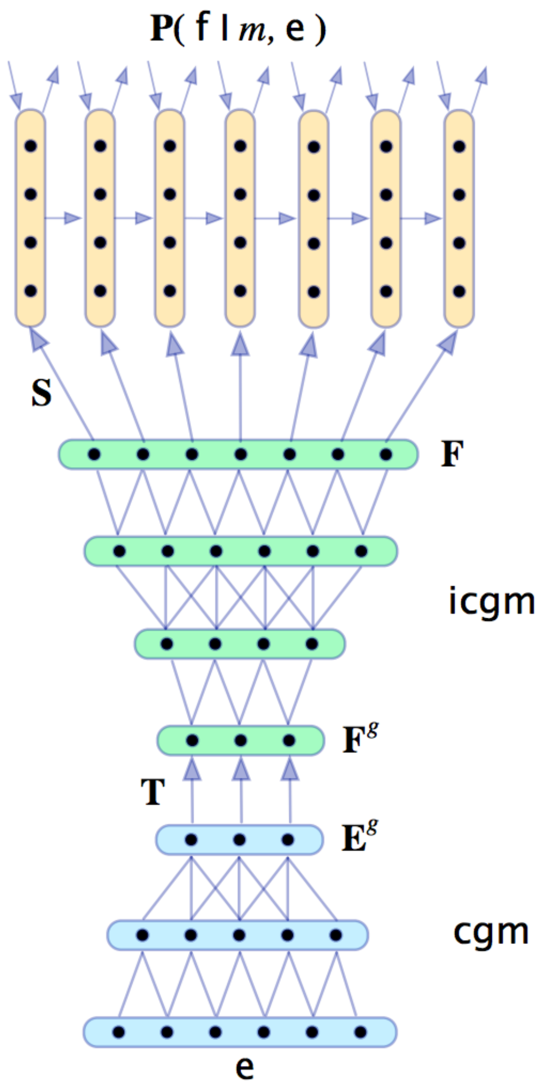
Convolutional Neural Networks (Kalchbrenner and Blunsom, 2013)



- Build sentence representation bottom-up
  - merge any n neighboring nodes
  - n may be 2, 3, ...

- Generate target sentence by inverting the process

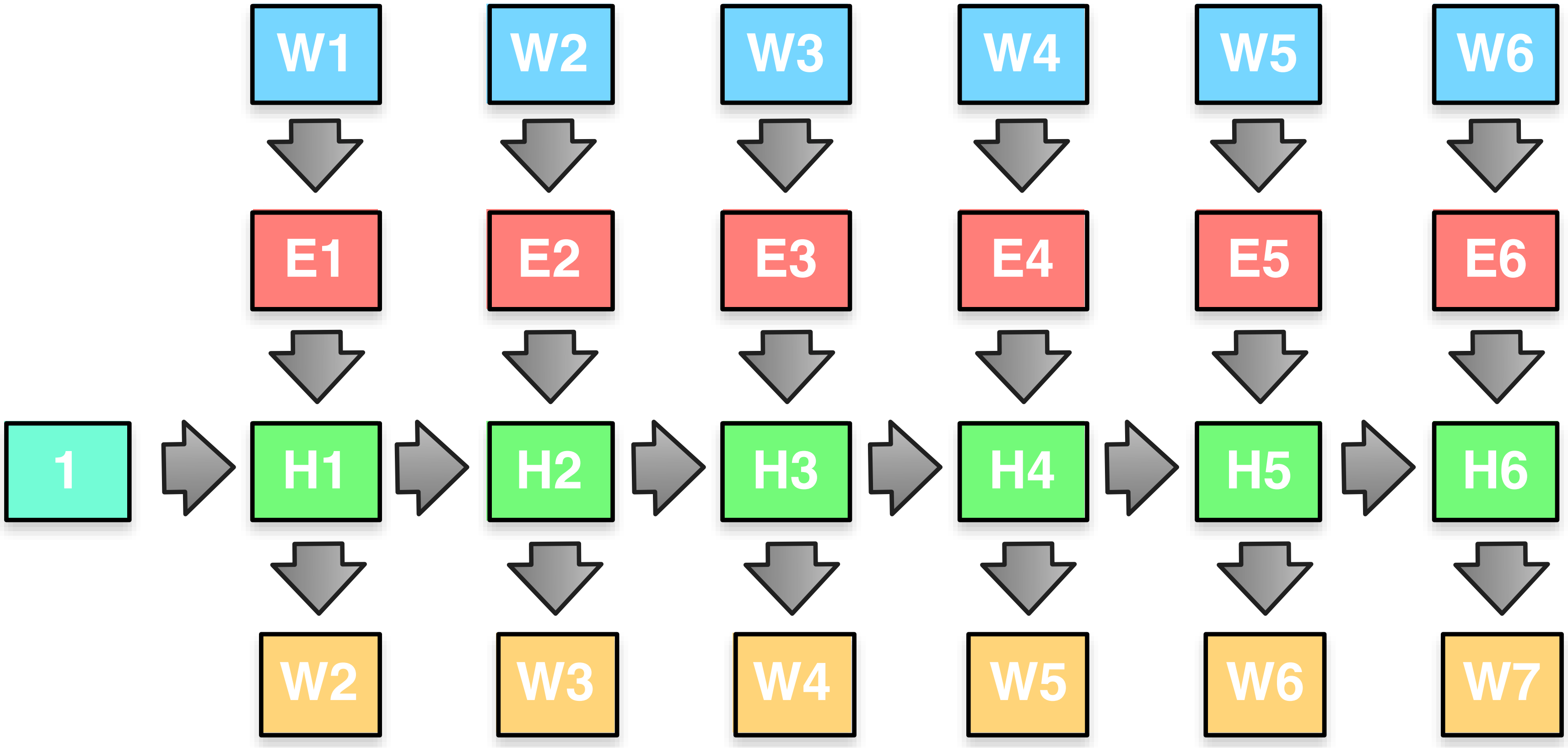
Generation



- Encode with convolutional neural network
- Decode with convolutional neural network
- Also include a linear recurrent neural network
- Important: predict length of output sentence
- Does it work?
  used successfully in re-ranking (Cho et al., 2014)

neural translation with alignment model

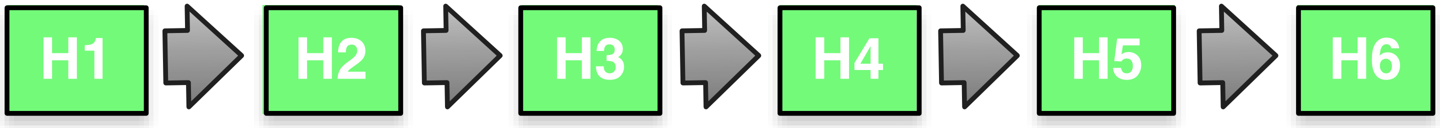
Some Preparation



- Train a recurrent neural network language model on the source side

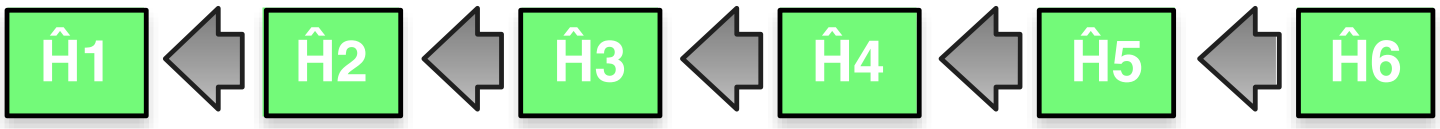
Hidden Language Model States

- This gives us the hidden states



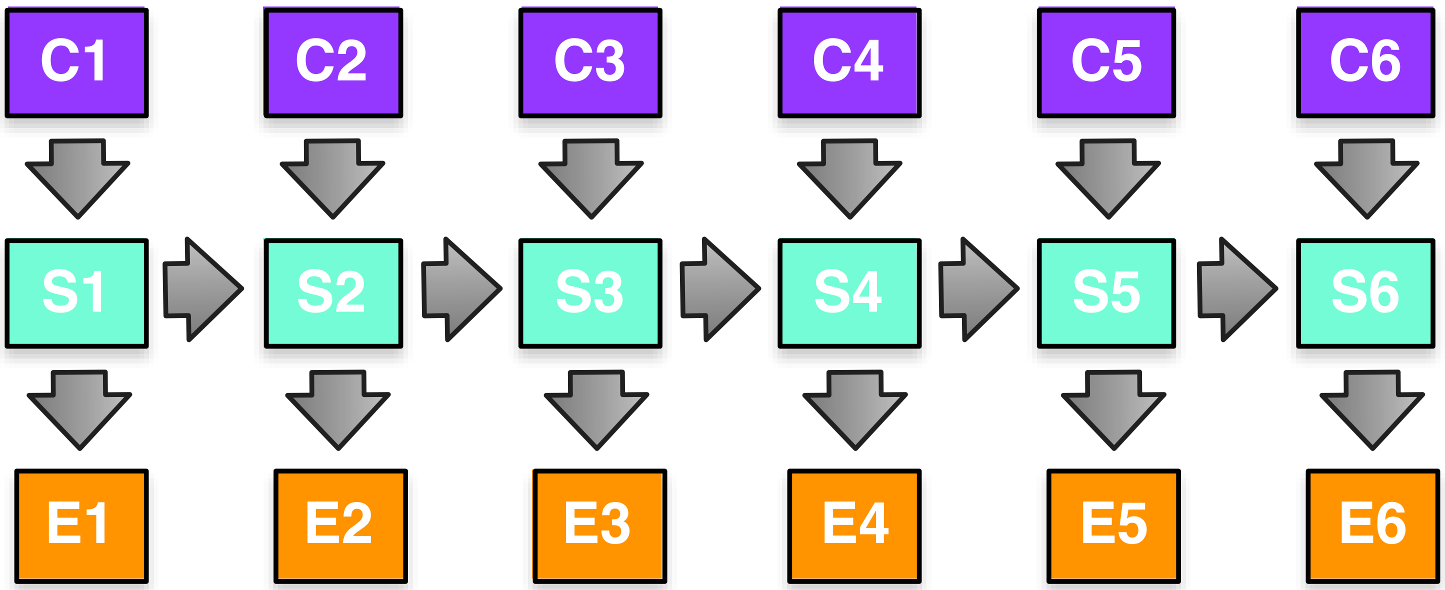
- These encode left context for each word

- Same process in reverse: right context for each word



Translation Model

- We want to have a recurrent neural network predicting output words e_i



- Somehow informed by the source context c_i, specific to each output word

Alignment Model

- Given
  - the previous state of the target RNN s_{i-1}
  - the representation of any source word $h_j = (\overleftarrow{h_j}, \overrightarrow{h_j})$

- Predict an alignment probability $a(s_{i-1}, h_j)$
  (of course, model with with a neural network)

- Normalize (softmax)

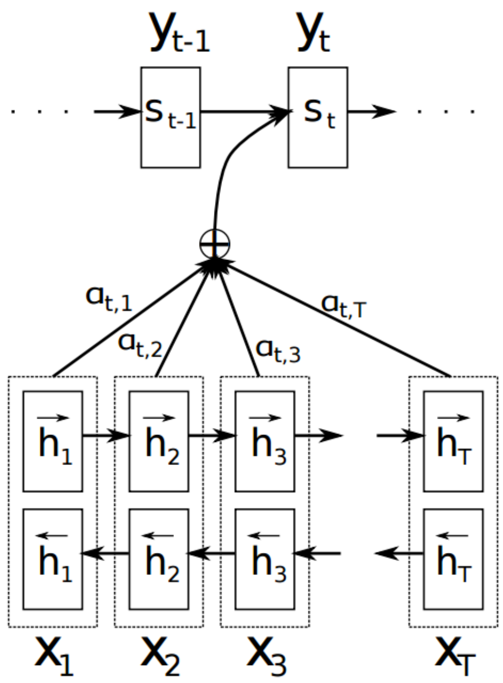
$$a_{ij} = \frac{\exp(a(s_{i-1}, h_j))}{\sum_k \exp(a(s_{i-1}, h_k))}$$

- Relevant source context: average weight of input word representations

$$c_i = \sum_j a_{ij} h_j$$

Bahdanau, Cho, and Bengio (2015)

- Putting it all together



- Model jointly trained to align and translate

conclusions

Incremental: Neural Components

- Modelling existing components with neural networks, e.g.,
  - language model
  - phrase translation
  - reordering model

- Conditional probability distribution $\rightarrow$ feed forward neural network

- Sequence model $\rightarrow$ recurrent neural network

- Neural networks allow integration of richer context
  - may cause problems for decoding (state splitting)

  $\rightarrow$ use only in re-ranking

Leap Forward: Neural Machine Translation

- No more beam search: hidden state capture all ambiguity
- But: proposed models feel like
  - IBM Model 1: condition broadly on the source sentence
  - IBM Model 2: use of a word-based alignment model
- It is a long climb to more structure in the model (phrases, syntax)...