# Machine Translation

Philipp Koehn

30 April 2019

# Machine Translation: French



Obama et Romney prévoient de mener campagne dans les «swing states» à un rythme effréné pour les quatre derniers jours avant l'élection. L'Ohio se présente comme l'Etat le plus disputé du pays.

Obama and Romney plan to campaign in the "swing states" at a breakneck pace for the last four days before the election. The Ohio State presents itself as the most played country.

# No Single Right Answer

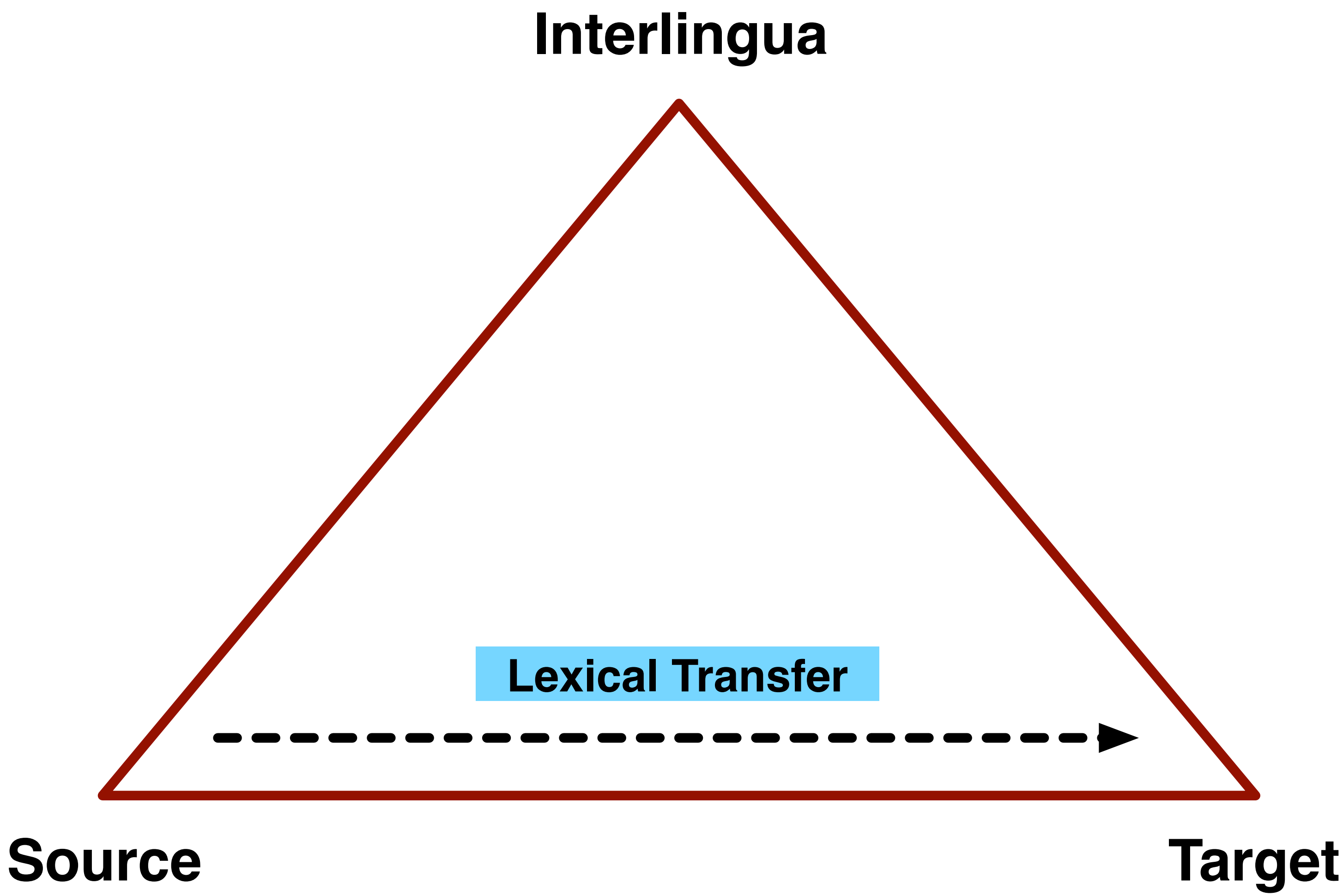


这个 机场 的 安全 工作 由 以色列 方面 负责 .

Israeli officials are responsible for airport security.
Israel is in charge of the security at this airport.
The security work for this airport is the responsibility of the Israel government.
Israeli side was in charge of the security of this airport.
Israel is responsible for the airport's security.
Israel is responsible for safety work at this airport.
Israel presides over the security of the airport.
Israel took charge of the airport security.
The safety of this airport is taken charge of by Israel.
This airport's security is the responsibility of the Israeli security officials.

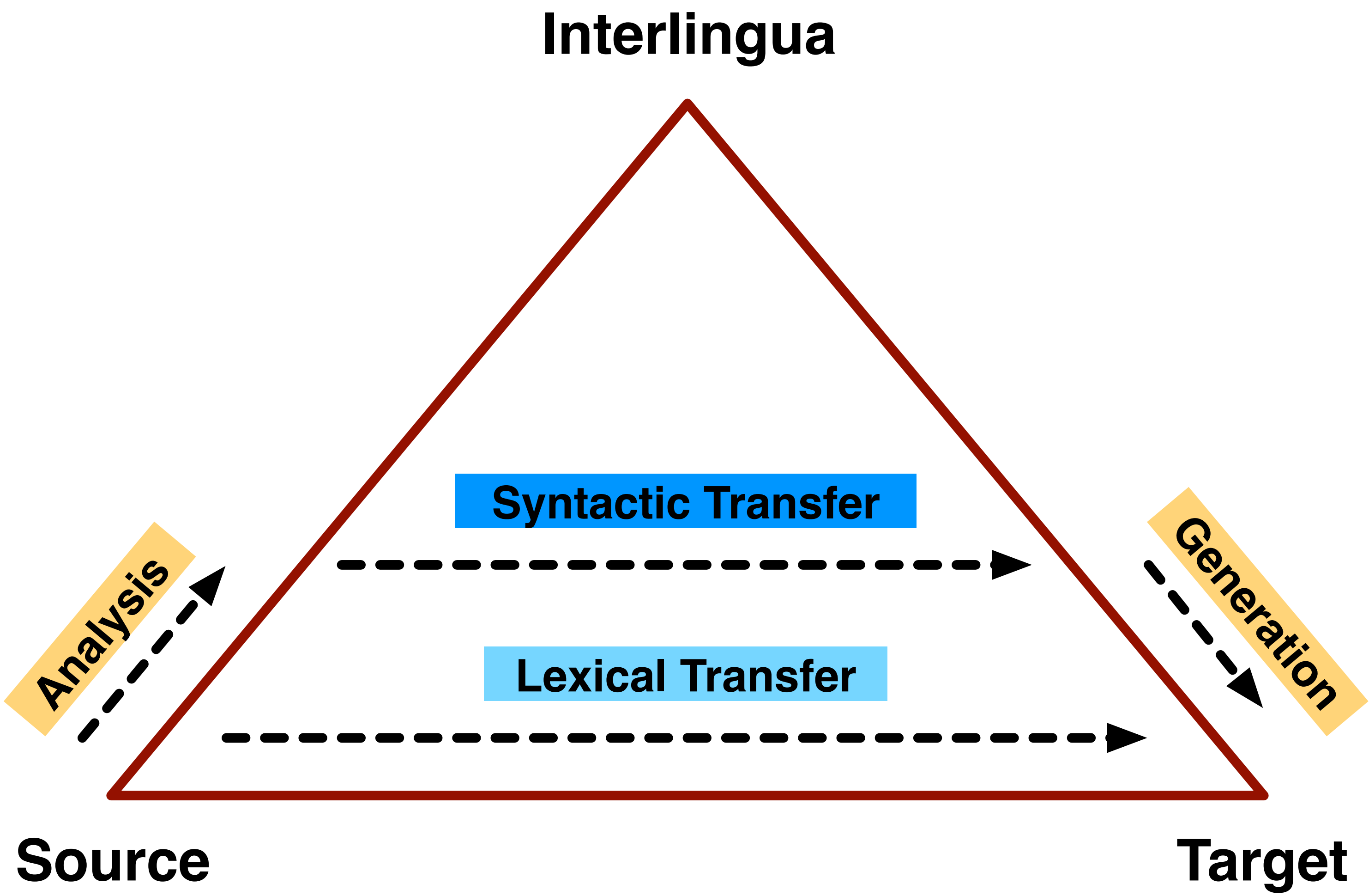
# A Clear Plan





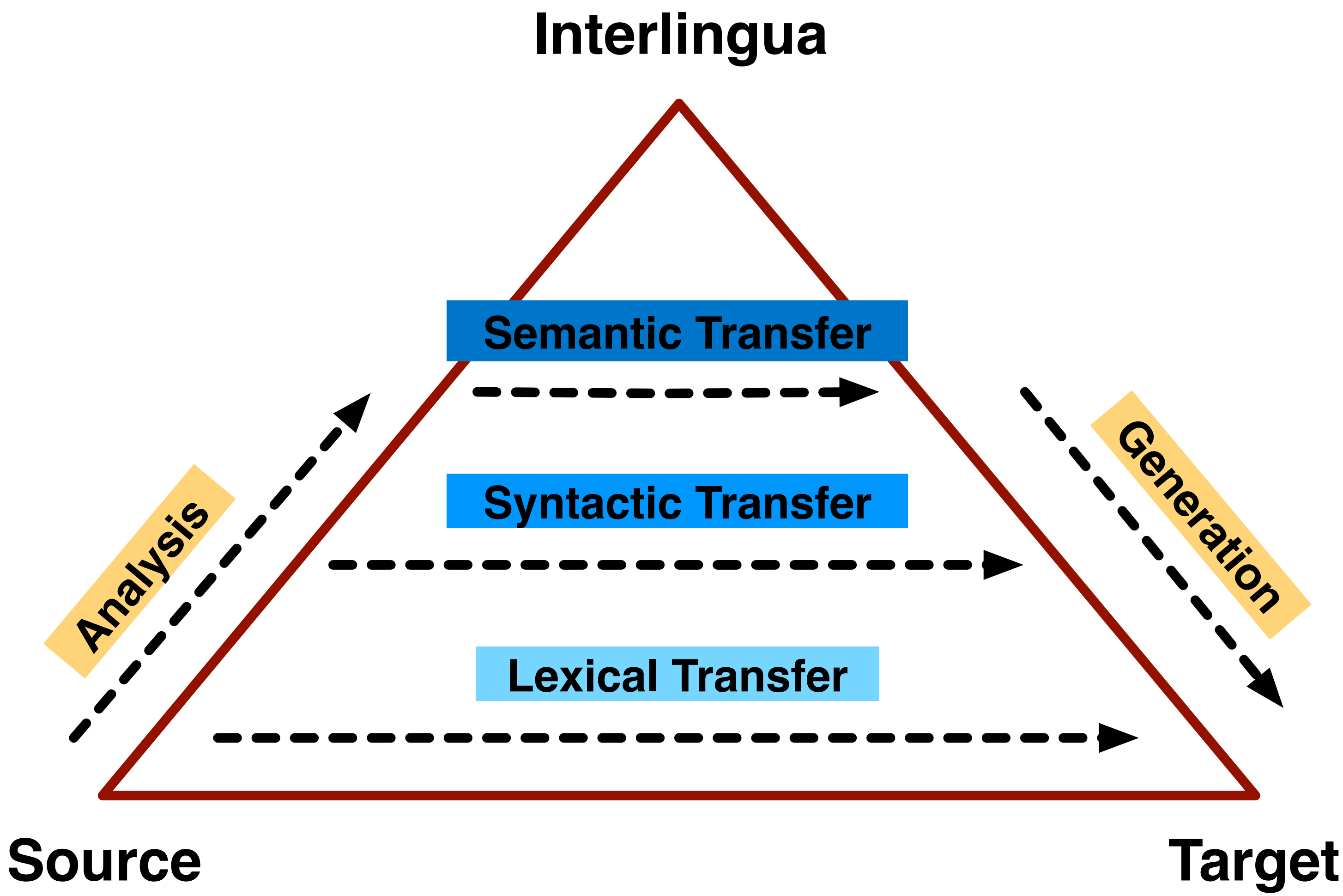
# A Clear Plan



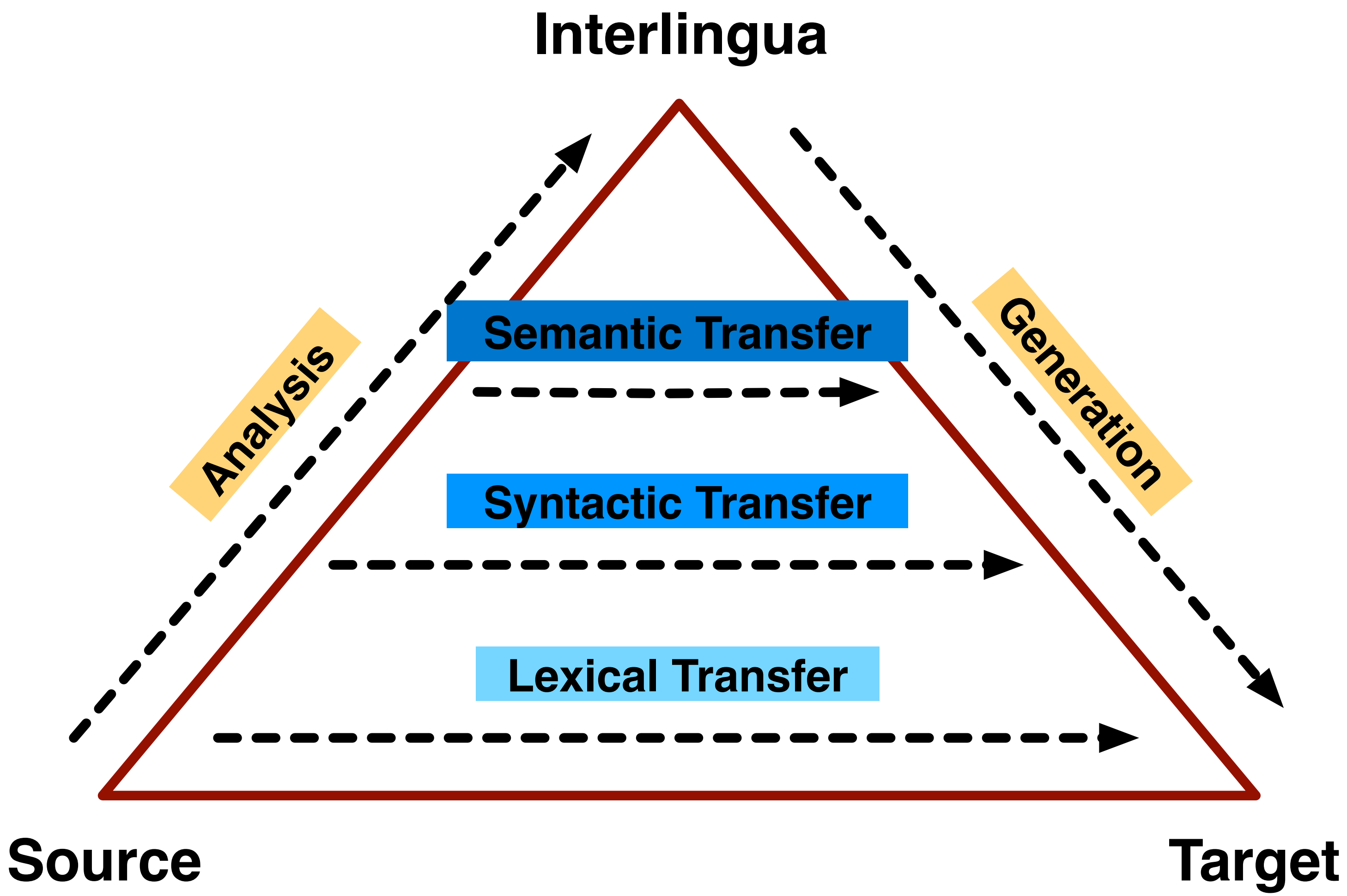

Interlingua
Syntactic Transfer
Lexical Transfer
Analysis
Generation
Source
Target

# A Clear Plan



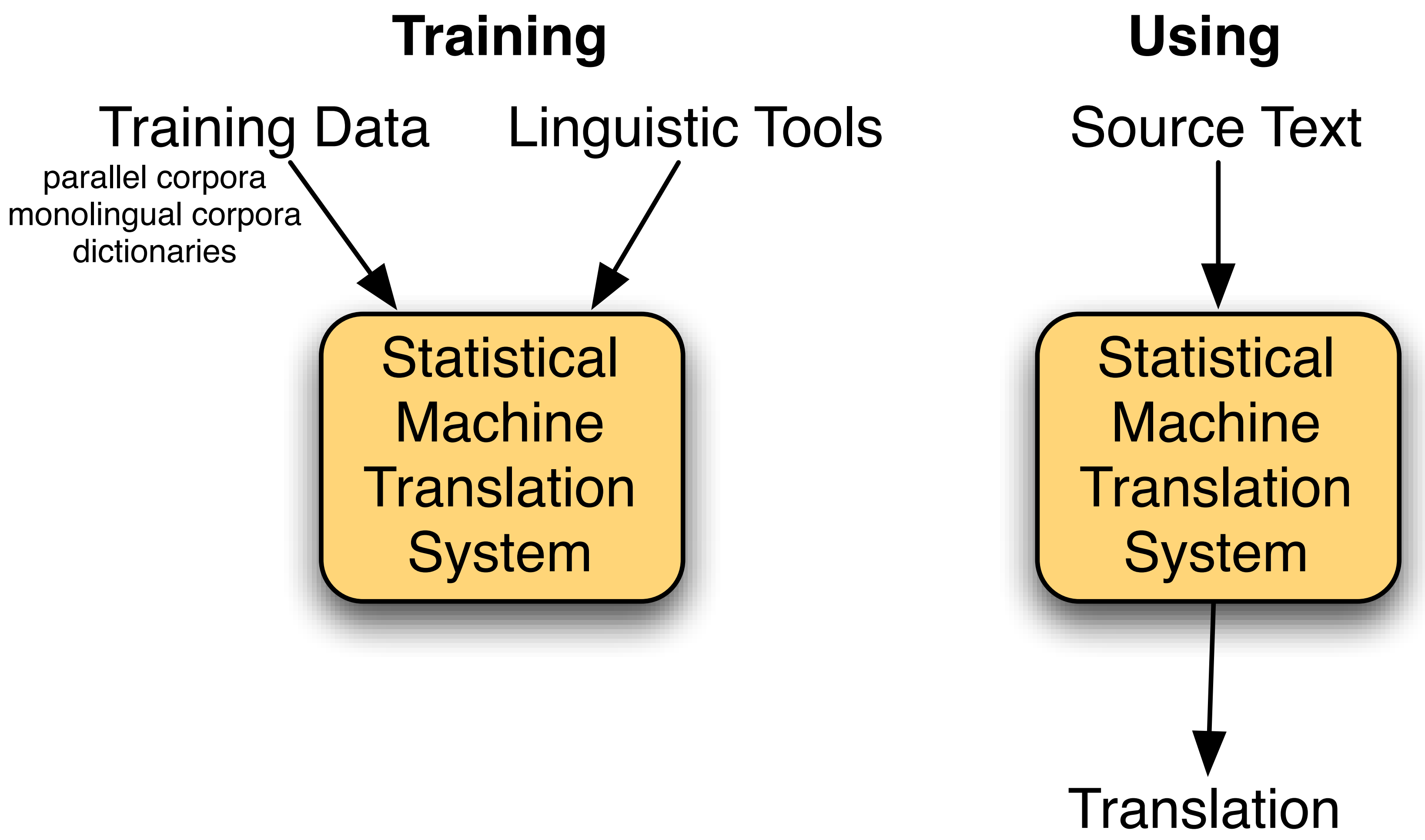

Interlingua
Semantic Transfer
Syntactic Transfer
Lexical Transfer
Analysis
Generation
Source
Target

# A Clear Plan


Interlingua
Analysis
Generation
Semantic Transfer
Syntactic Transfer
Lexical Transfer
Source
Target

# Learning from Data





# why is that a good plan?

# Word Translation Problems



- Words are ambiguous

He deposited money in a bank account
with a high interest rate.

Sitting on the bank of the Mississippi,
a passing ship piqued his interest.

- How do we find the right meaning, and thus translation?

- Context should be helpful

# Syntactic Translation Problems



- Languages have different sentence structure

| das | behaupten | sie | wenigstens |
|---|---|---|---|
| this | claim | they | at least |
| the | | she | |

- Convert from object-verb-subject (OVS) to subject-verb-object (SVO)

- Ambiguities can be resolved through syntactic analysis
  - the meaning the of das not possible (not a noun phrase)
  - the meaning she of sie not possible (subject-verb agreement)

# Semantic Translation Problems



- Pronominal anaphora

I saw the movie and it is good.

- How to translate it into German (or French)?
  - it refers to movie
  - movie translates to Film
  - Film has masculine gender
  - ergo: it must be translated into masculine pronoun er

- We are not handling this very well [Le Nagard and Koehn, 2010]

# Semantic Translation Problems

- Coreference

Whenever I visit my uncle and his daughters,
I can't decide who is my favorite cousin.

- How to translate cousin into German? Male or female?

- Complex inference required

# Semantic Translation Problems



- Discourse

Since you brought it up, I do not agree with you.

Since you brought it up, we have been working on it.

- How to translated since? Temporal or conditional?

- Analysis of discourse structure — a hard problem

# Learning from Data



- What is the best translation?

Sicherheit → security
Sicherheit → safety
Sicherheit → certainty

# Learning from Data



- What is the best translation?

Sicherheit → security 14,516
Sicherheit → safety 10,015
Sicherheit → certainty 334

- Counts in European Parliament corpus

# Learning from Data

- What is the best translation?

Sicherheit → security 14,516
Sicherheit → safety 10,015
Sicherheit → certainty 334

- Phrasal rules

Sicherheitspolitik → security policy 1580
Sicherheitspolitik → safety policy 13
Sicherheitspolitik → certainty policy 0

Lebensmittelsicherheit → food security 51
Lebensmittelsicherheit → food safety 1084
Lebensmittelsicherheit → food certainty 0

Rechtssicherheit → legal security 156
Rechtssicherheit → legal safety 5
Rechtssicherheit → legal certainty 723

# Learning from Data



- What is most fluent?

a problem for translation

a problem of translation

a problem in translation

# Learning from Data



- What is most fluent?

a problem for translation 13,000

a problem of translation 61,600

a problem in translation 81,700

- Hits on Google

# Learning from Data



- What is most fluent?

a problem for translation 13,000

a problem of translation 61,600

a problem in translation 81,700

a translation problem 235,000

# Learning from Data

- What is most fluent?

police disrupted the demonstration
police broke up the demonstration
police dispersed the demonstration
police ended the demonstration
police dissolved the demonstration
police stopped the demonstration
police suppressed the demonstration
police shut down the demonstration

# Learning from Data



- What is most fluent?

police disrupted the demonstration 2,140

police broke up the demonstration 66,600

police dispersed the demonstration 25,800

police ended the demonstration 762

police dissolved the demonstration 2,030

police stopped the demonstration 722,000

police suppressed the demonstration 1,400

police shut down the demonstration 2,040

# word alignment

# Lexical Translation



- How to translate a word → look up in dictionary

  **Haus** — house, building, home, household, shell.

- Multiple translations
  - some more frequent than others
  - for instance: house, and building most common
  - special cases: Haus of a snail is its shell

- Note: In all lectures, we translate from a foreign language into English

# Collect Statistics



Look at a parallel corpus (German text along with English translation)

| **Translation of *Haus*** | **Count** |
|---|---|
| house | 8,000 |
| building | 1,600 |
| home | 200 |
| household | 150 |
| shell | 50 |

# Estimate Translation Probabilities

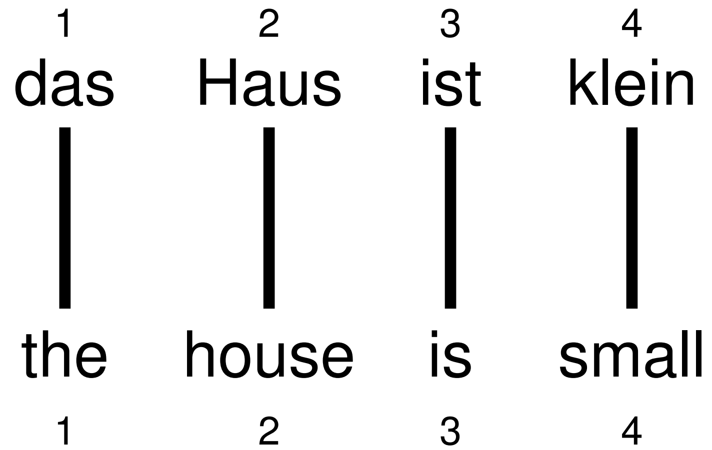


Maximum likelihood estimation

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

# Alignment

- In a parallel text (or when we translate), we align words in one language with the words in the other




- Word positions are numbered 1–4

# Alignment Function



- Formalizing alignment with an alignment function

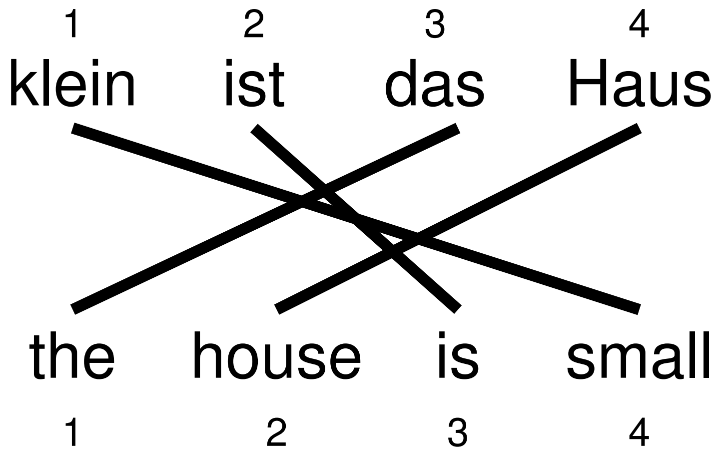
- Mapping an English target word at position $i$ to a German source word at position $j$ with a function $a : i \rightarrow j$

- Example

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

# Reordering

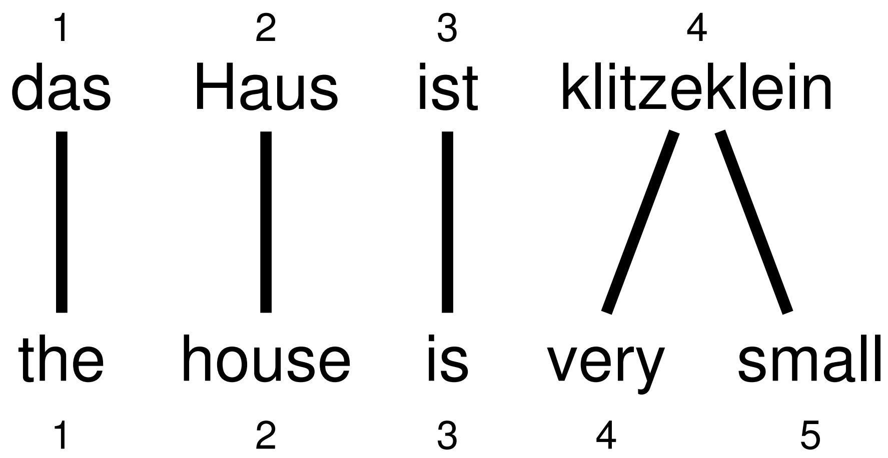
Words may be reordered during translation




$$a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$

# One-to-Many Translation

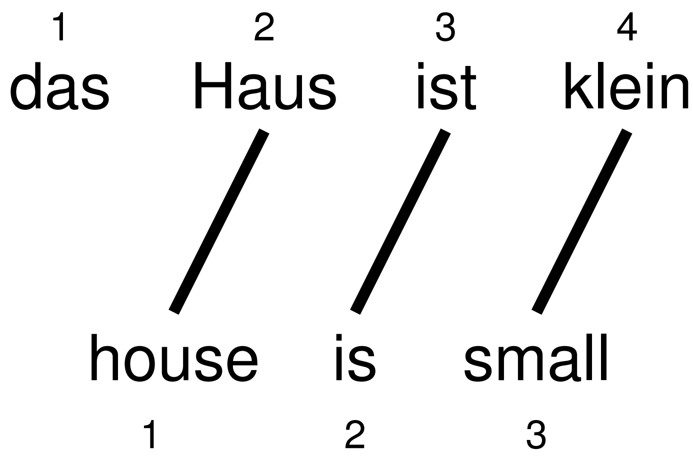
A source word may translate into multiple target words




$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

# Dropping Words

Words may be dropped when translated
(German article das is dropped)

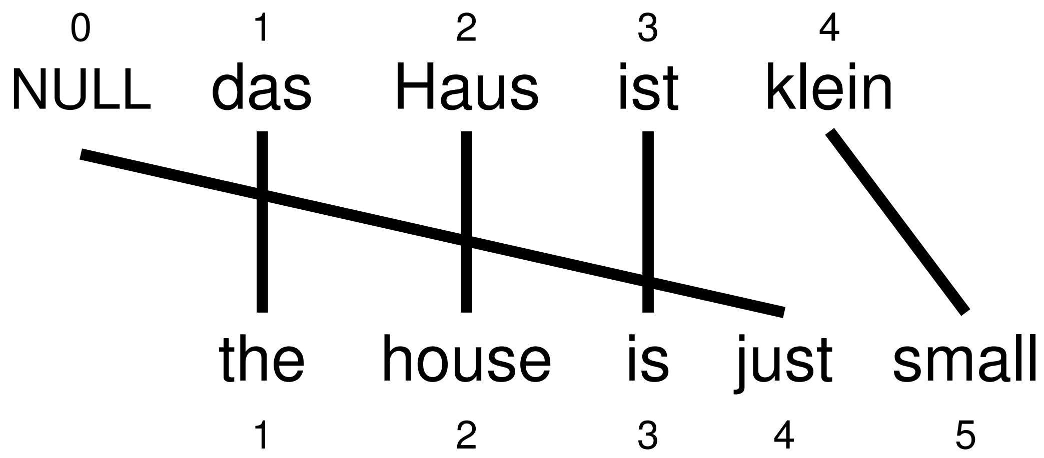



$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

# Inserting Words



- Words may be added during translation
  - The English just does not have an equivalent in German
  - We still need to map it to something: special NULL token




$$a: \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

# IBM Model 1



- Generative model: break up translation process into smaller steps
  - IBM Model 1 only uses lexical translation
- Translation probability
  - for a foreign sentence $\mathbf{f} = (f_1, ..., f_{l_f})$ of length $l_f$
  - to an English sentence $\mathbf{e} = (e_1, ..., e_{l_e})$ of length $l_e$
  - with an alignment of each English word $e_j$ to a foreign word $f_i$ according to the alignment function $a : j \rightarrow i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

  - parameter $\epsilon$ is a normalization constant

# Example

das

| $e$ | $t(e\|f)$ |
|---|---|
| the | 0.7 |
| that | 0.15 |
| which | 0.075 |
| who | 0.05 |
| this | 0.025 |

Haus

| $e$ | $t(e\|f)$ |
|---|---|
| house | 0.8 |
| building | 0.16 |
| home | 0.02 |
| household | 0.015 |
| shell | 0.005 |

ist

| $e$ | $t(e\|f)$ |
|---|---|
| is | 0.8 |
| 's | 0.16 |
| exists | 0.02 |
| has | 0.015 |
| are | 0.005 |

klein

| $e$ | $t(e\|f)$ |
|---|---|
| small | 0.4 |
| little | 0.4 |
| short | 0.1 |
| minor | 0.06 |
| petty | 0.04 |

$$
\begin{aligned}
p(e,a|f) &= \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\
&= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\
&= 0.0028\epsilon
\end{aligned}
$$

# em algorithm

# Learning Lexical Translation Models

- We would like to estimate the lexical translation probabilities $t(e|f)$ from a parallel corpus

- ... but we do not have the alignments

- Chicken and egg problem
  - if we had the *alignments*,
    → we could estimate the *parameters* of our generative model
  - if we had the *parameters*,
    → we could estimate the *alignments*

# EM Algorithm

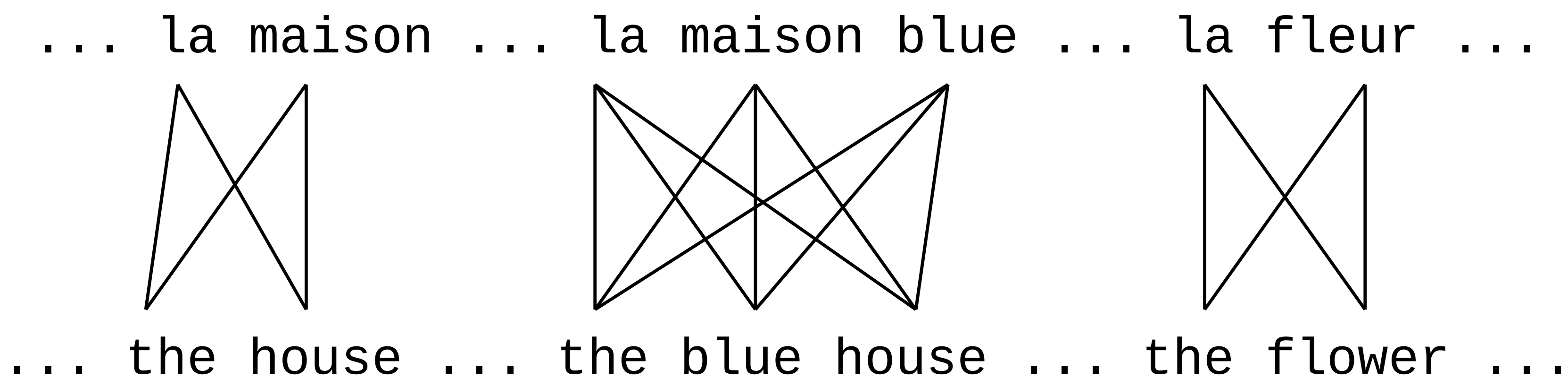


- Incomplete data
  - if we had *complete data,* would could estimate *model*
  - if we had *model,* we could fill in the *gaps in the data*

- Expectation Maximization (EM) in a nutshell
  1. initialize model parameters (e.g. uniform)
  2. assign probabilities to the missing data
  3. estimate model parameters from completed data
  4. iterate steps 2–3 until convergence

# EM Algorithm

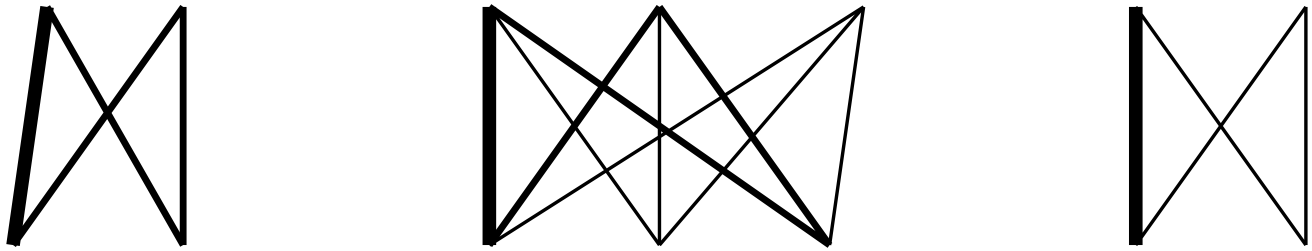





- Initial step: all alignments equally likely
- Model learns that, e.g., la is often aligned with the

# EM Algorithm

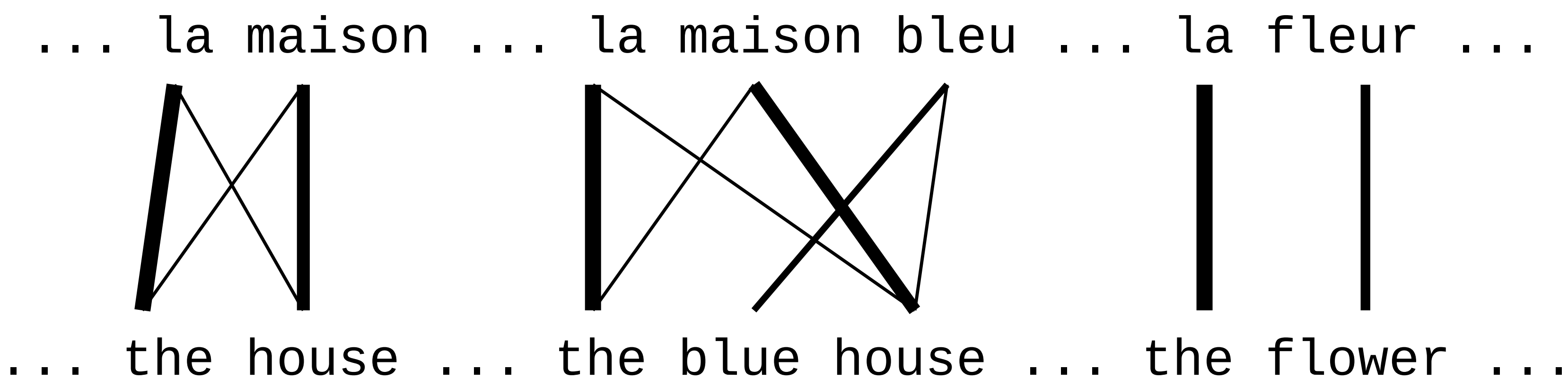
... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

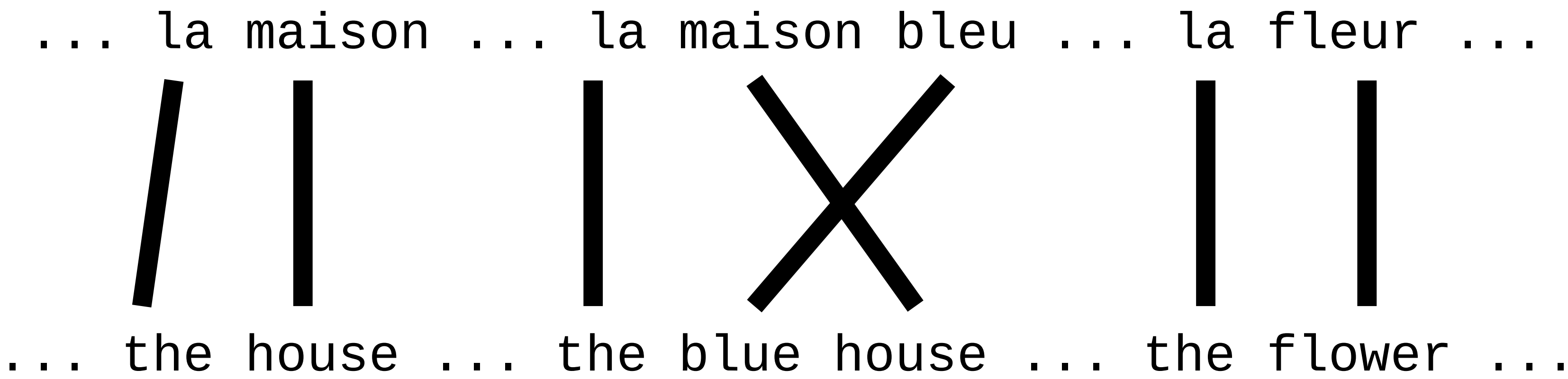
- After one iteration
- Alignments, e.g., between la and the are more likely

# EM Algorithm



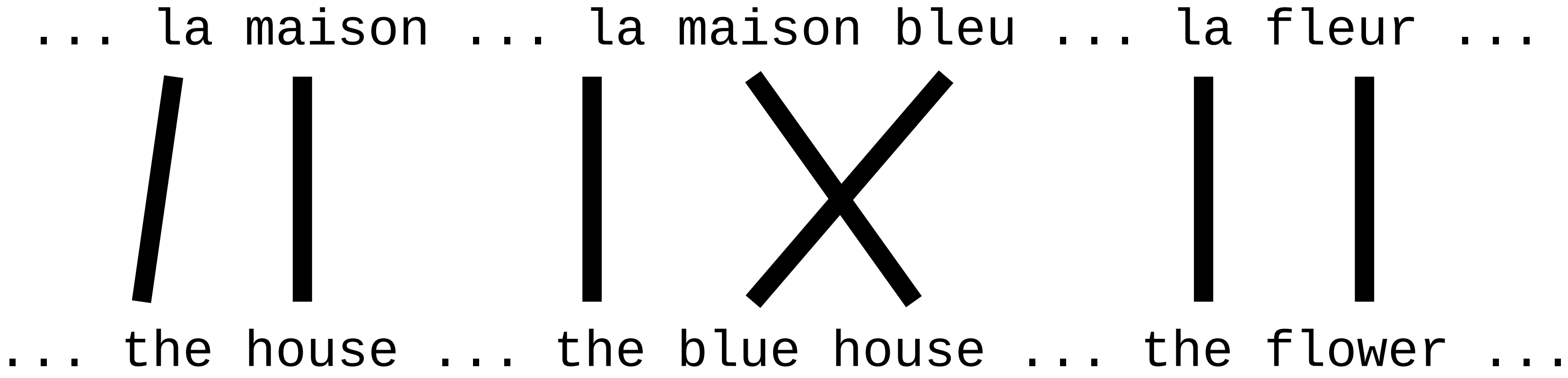

- After another iteration
- It becomes apparent that alignments, e.g., between fleur and flower are more likely (pigeon hole principle)

# EM Algorithm




- Convergence
- Inherent hidden structure revealed by EM

# EM Algorithm




```
p(la|the) = 0.453
p(le|the) = 0.334
p(maison|house) = 0.876
p(bleu|blue) = 0.563
...
```

- Parameter estimation from the aligned corpus

# IBM Model 1 and EM

- EM Algorithm consists of two steps

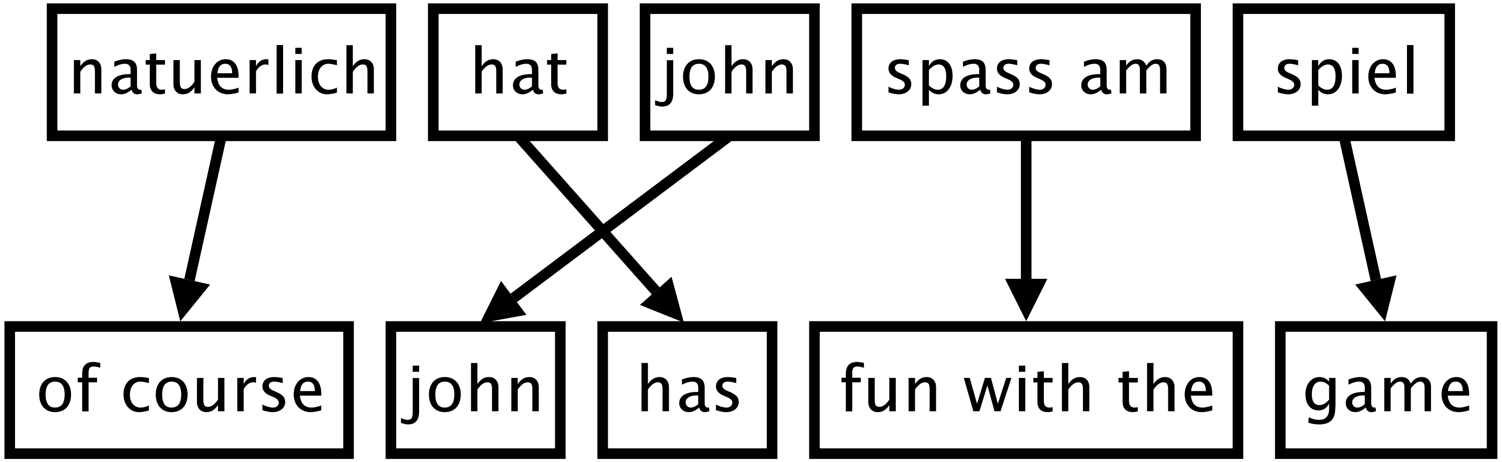
- Expectation-Step: Apply model to the data
  - parts of the model are hidden (here: alignments)
  - using the model, assign probabilities to possible values

- Maximization-Step: Estimate model from data
  - take assign values as fact
  - collect counts (weighted by probabilities)
  - estimate model from counts

- Iterate these steps until convergence

# phrase-based models

# Phrase-Based Model




- Foreign input is segmented in phrases
- Each phrase is translated into English
- Phrases are reordered

# Phrase Translation Table



- Main knowledge source: table with phrase translations and their probabilities
- Example: phrase translations for natuerlich

| Translation | Probability $\phi(\bar{e}\|\bar{f})$ |
|---|---|
| of course | 0.5 |
| naturally | 0.3 |
| of course , | 0.15 |
| , of course , | 0.05 |

# Real Example



- Phrase translations for den Vorschlag learned from the Europarl corpus:

| English | $\phi(\bar{e}\|\bar{f})$ | English | $\phi(\bar{e}\|\bar{f})$ |
|---|---|---|---|
| the proposal | 0.6227 | the suggestions | 0.0114 |
| 's proposal | 0.1068 | the proposed | 0.0114 |
| a proposal | 0.0341 | the motion | 0.0091 |
| the idea | 0.0250 | the idea of | 0.0091 |
| this proposal | 0.0227 | the proposal , | 0.0068 |
| proposal | 0.0205 | its proposal | 0.0068 |
| of the proposal | 0.0159 | it | 0.0068 |
| the proposals | 0.0159 | ... | ... |

  - lexical variation (proposal vs suggestions)
  - morphological variation (proposal vs proposals)
  - included function words (the, a, ...)
  - noise (it)

# decoding

# Decoding

- We have a mathematical model for translation

$$p(\mathbf{e}|\mathbf{f})$$

- Task of decoding: find the translation $\mathbf{e}_{\text{best}}$ with highest probability

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f})$$

- Two types of error
  - the most probable translation is bad → fix the model
  - search does not find the most probably translation → fix the search

- Decoding is evaluated by search error, not quality of translations (although these are often correlated)

# Translation Process



- Task: translate this sentence from German into English

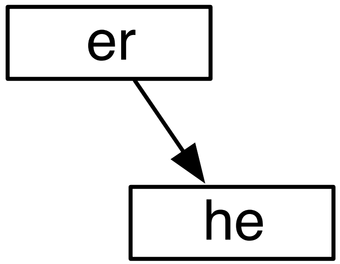
| | | | | | |
|---|---|---|---|---|---|
| **er** | **geht** | **ja** | **nicht** | **nach** | **hause** |

# Translation Process



- Task: translate this sentence from German into English

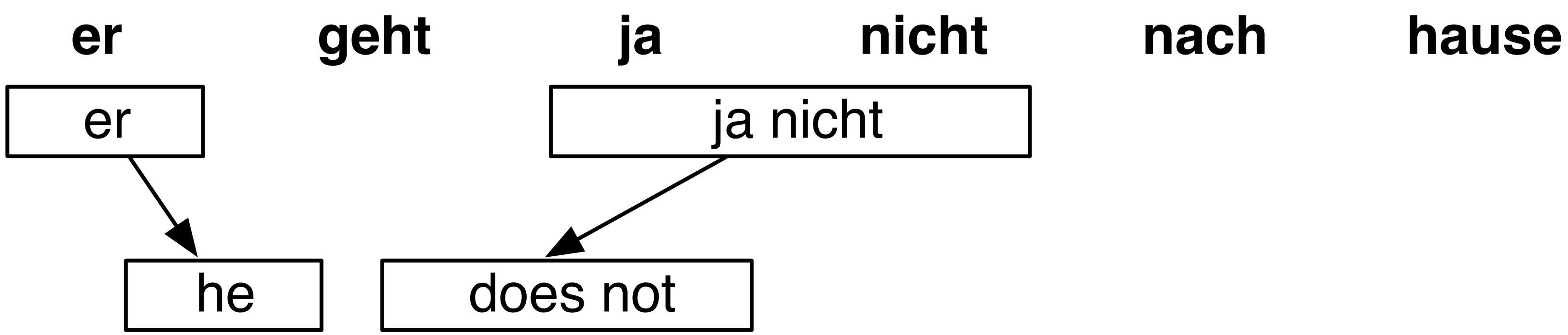
**er** **geht** **ja** **nicht** **nach** **hause**




- Pick phrase in input, translate

# Translation Process

- Task: translate this sentence from German into English

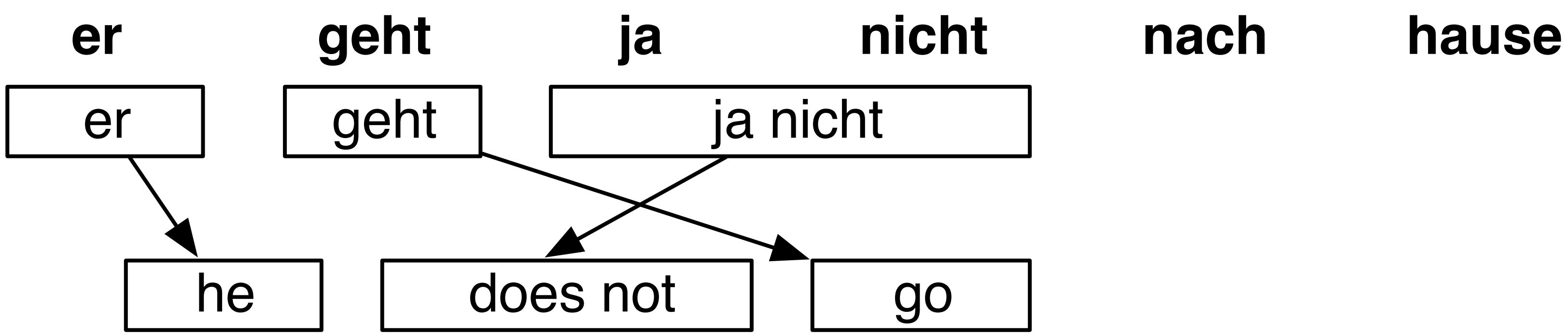



- Pick phrase in input, translate
  - it is allowed to pick words out of sequence reordering
  - phrases may have multiple words: many-to-many translation

# Translation Process



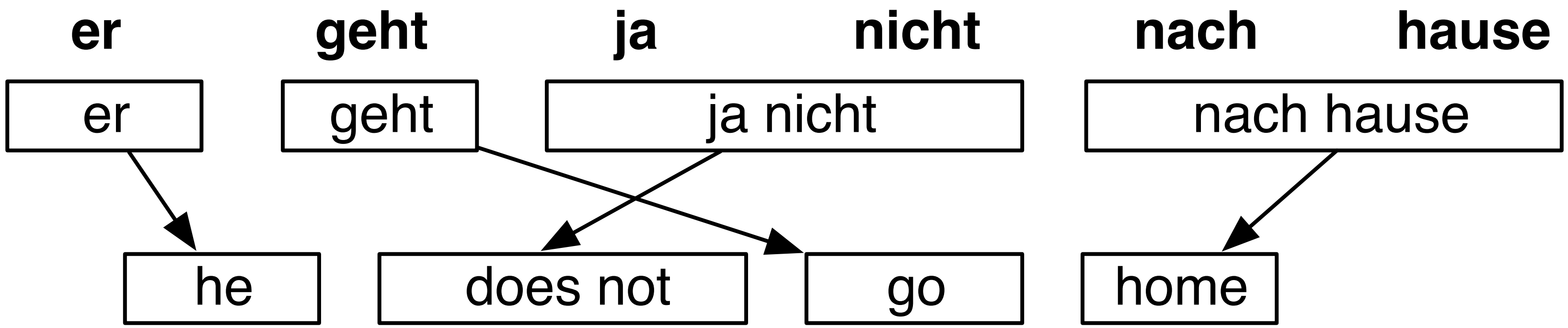
- Task: translate this sentence from German into English




- Pick phrase in input, translate

# Translation Process

- Task: translate this sentence from German into English

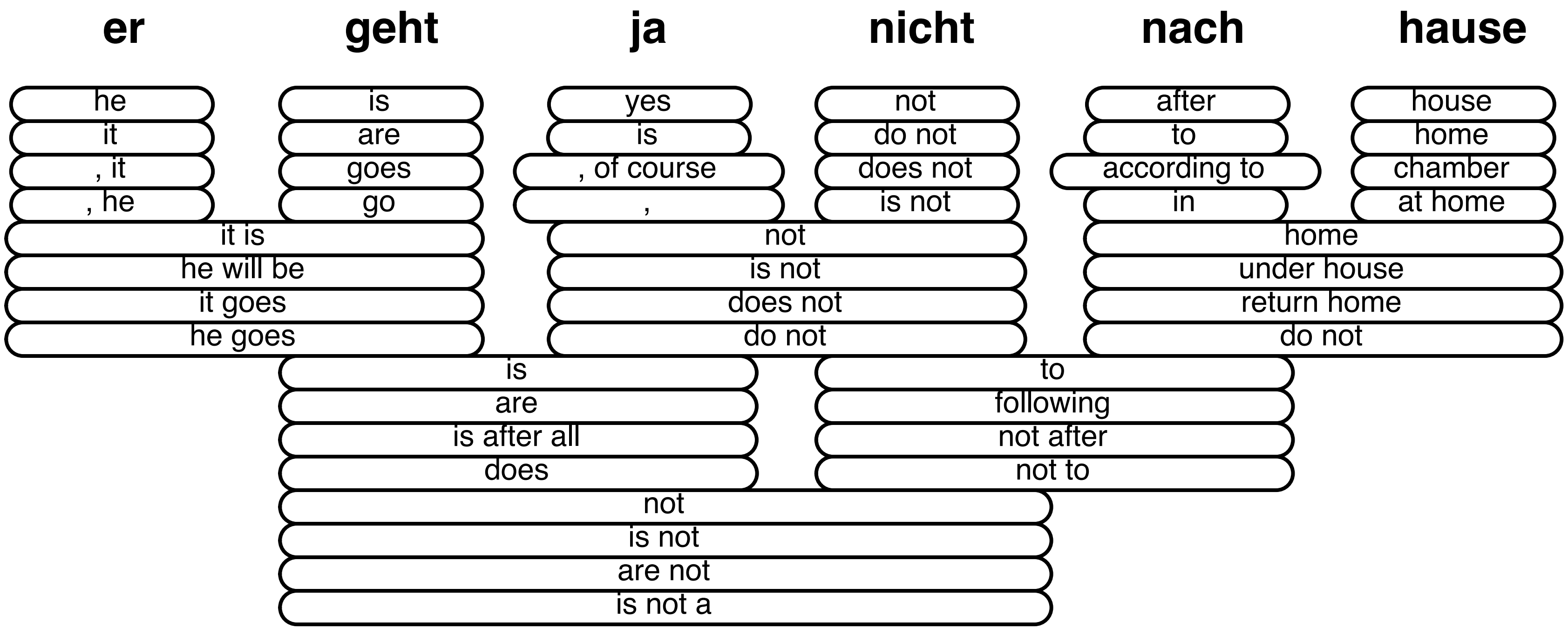



- Pick phrase in input, translate

# decoding process

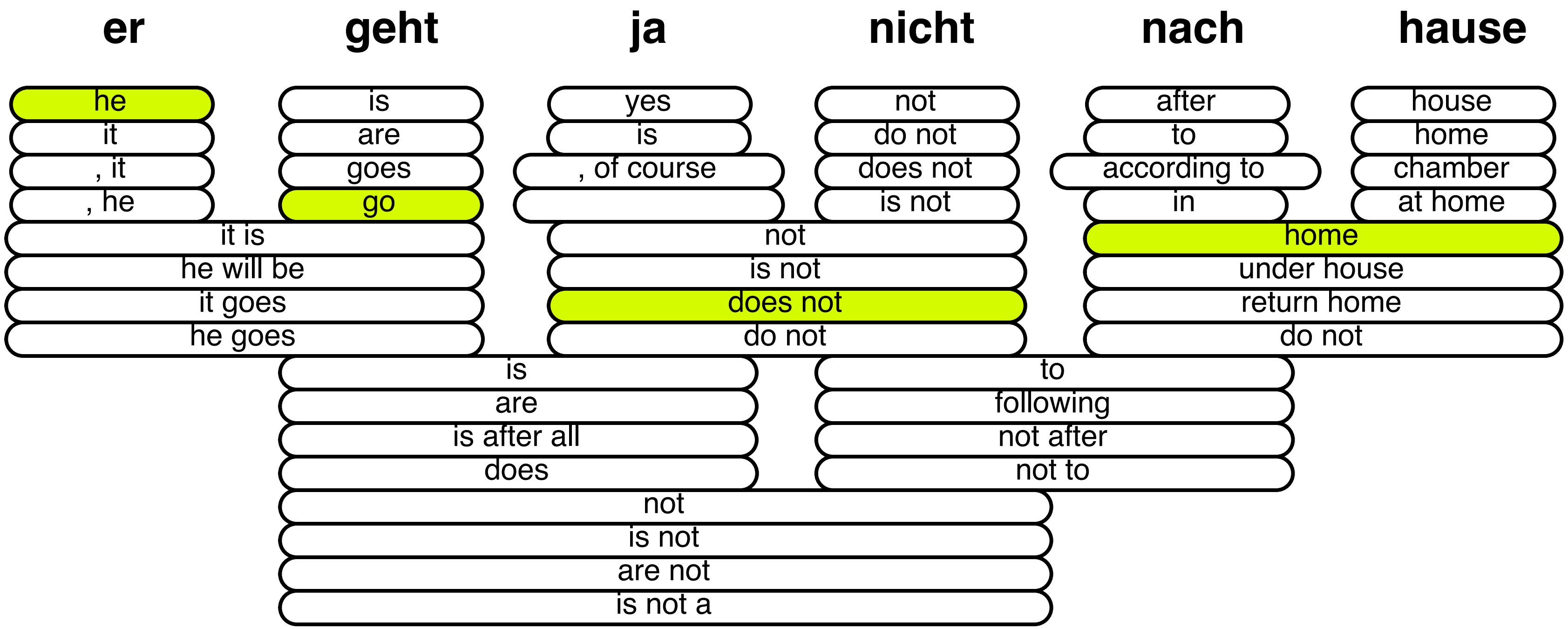
# Translation Options






- Many translation options to choose from
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain

# Translation Options

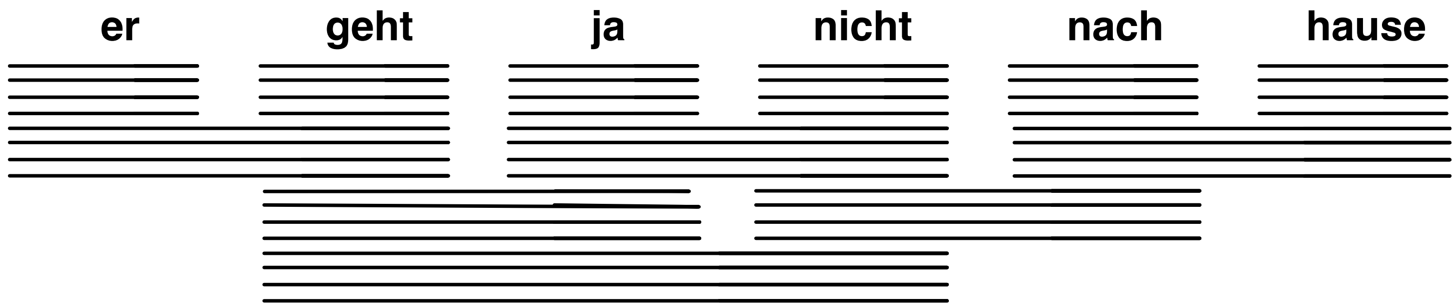



- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order

→ Search problem solved by heuristic beam search

# Decoding: Precompute Translation Options

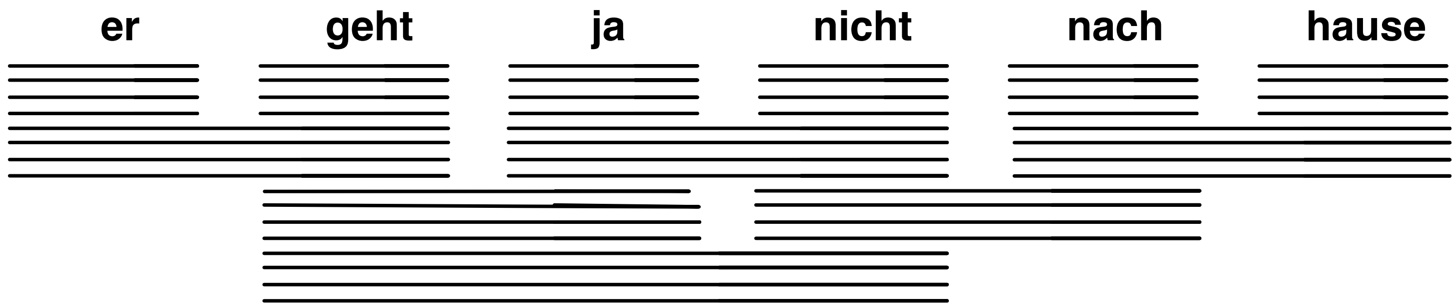


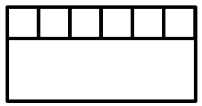



consult phrase translation table for all input phrases

# Decoding: Start with Initial Hypothesis

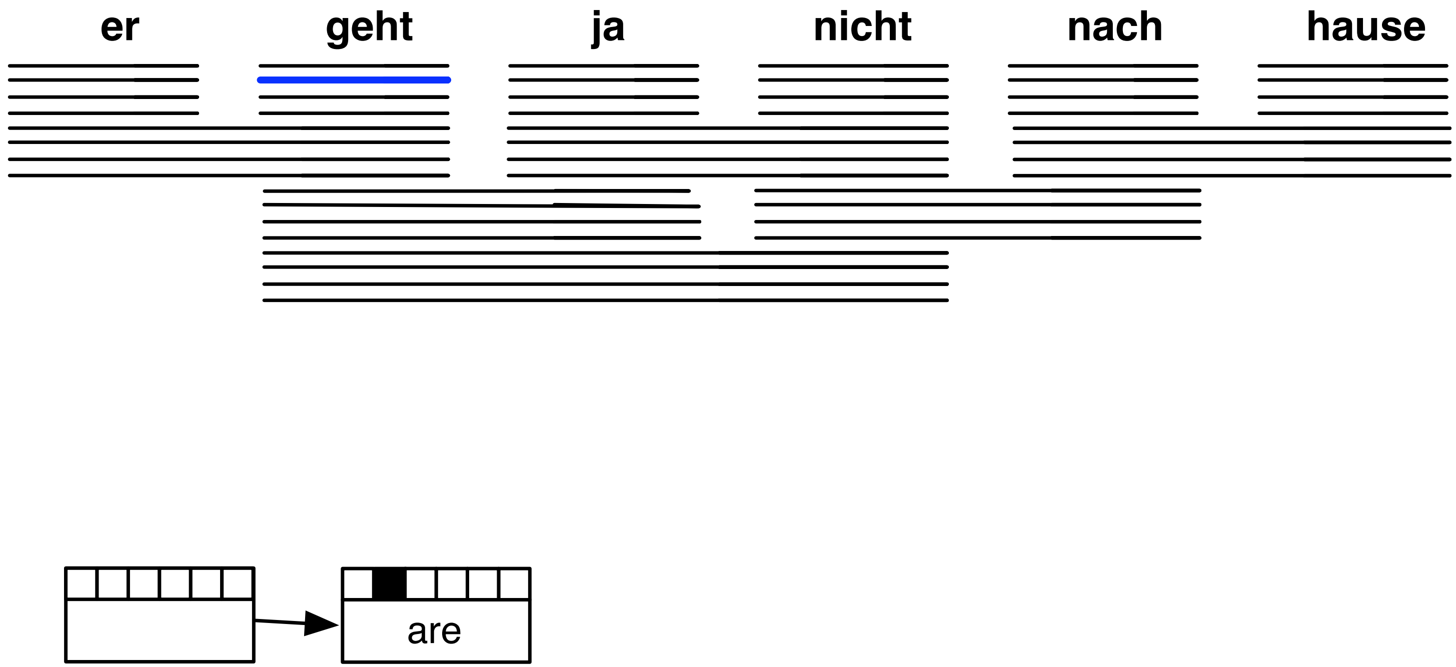




initial hypothesis: no input words covered, no output produced

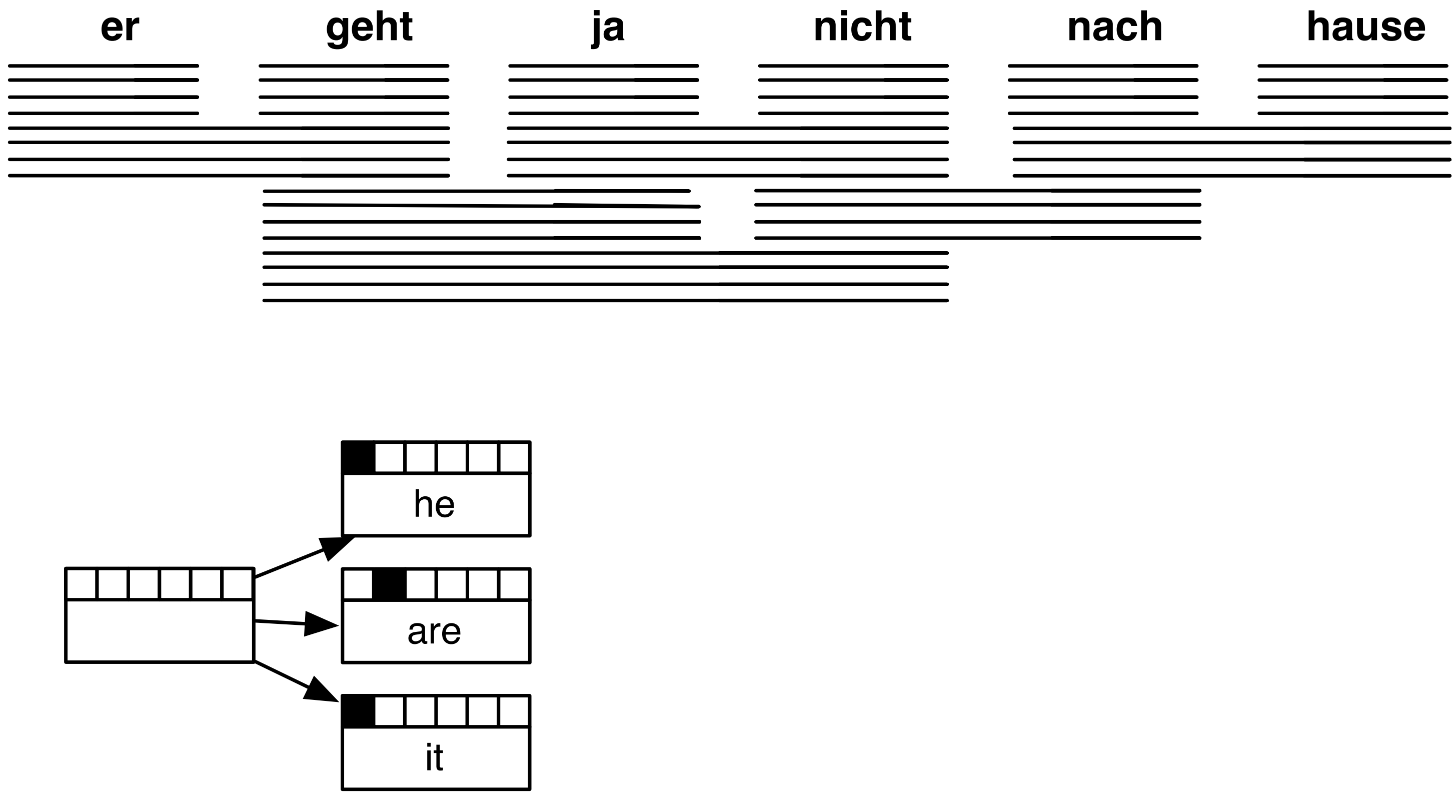
# Decoding: Hypothesis Expansion




pick any translation option, create new hypothesis

# Decoding: Hypothesis Expansion

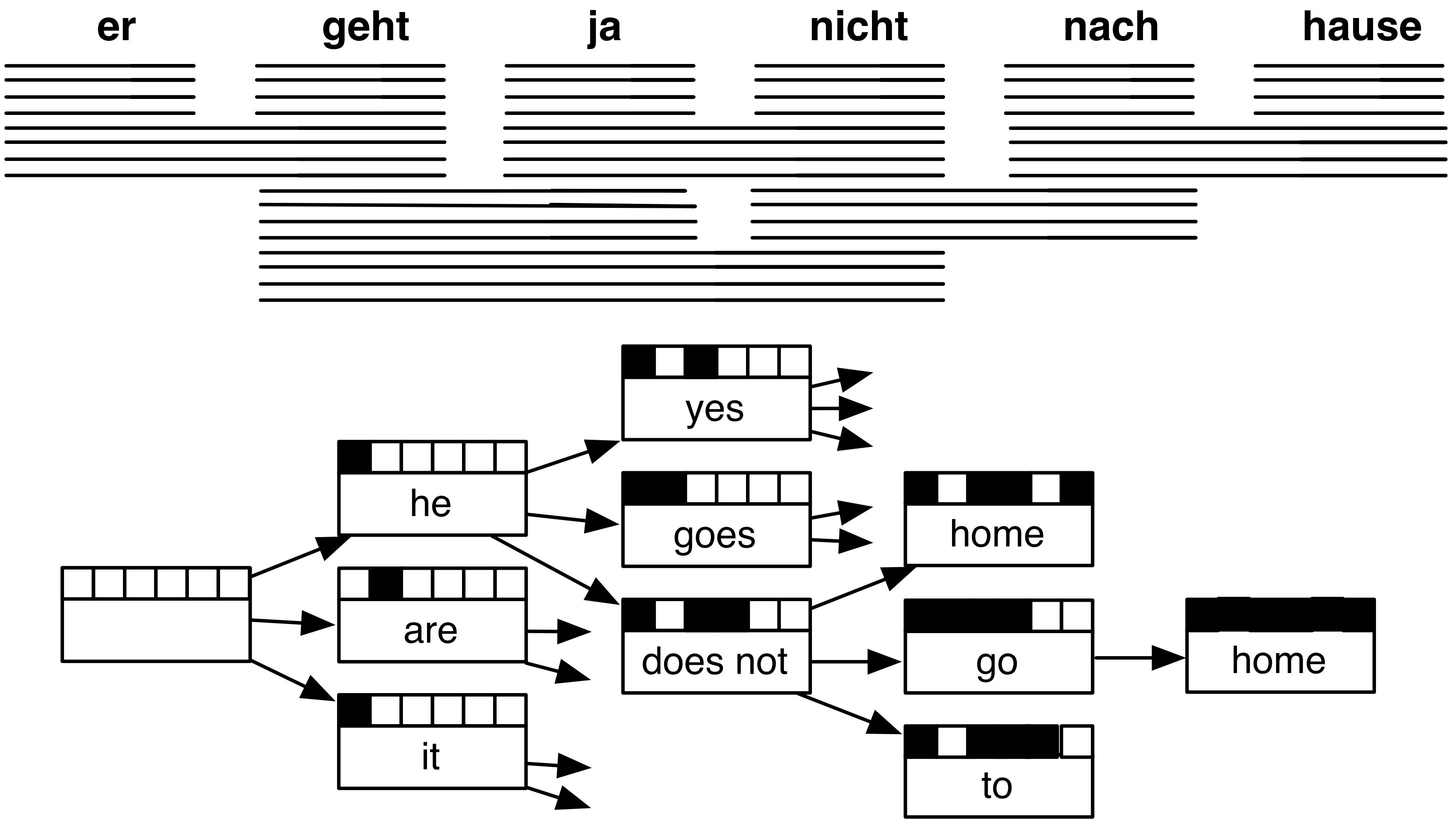





create hypotheses for all other translation options

# Decoding: Hypothesis Expansion

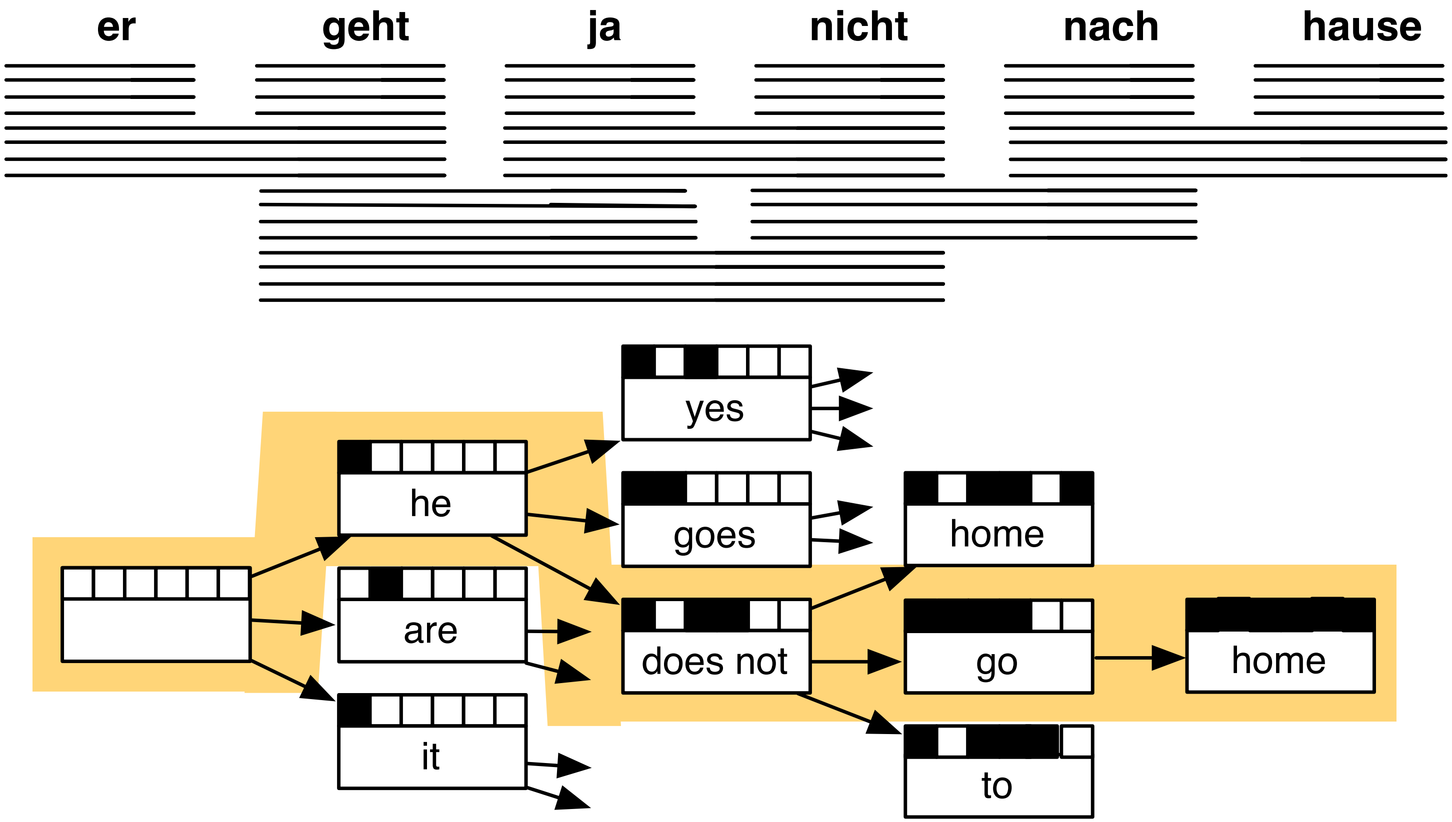





also create hypotheses from created partial hypothesis

# Decoding: Find Best Path

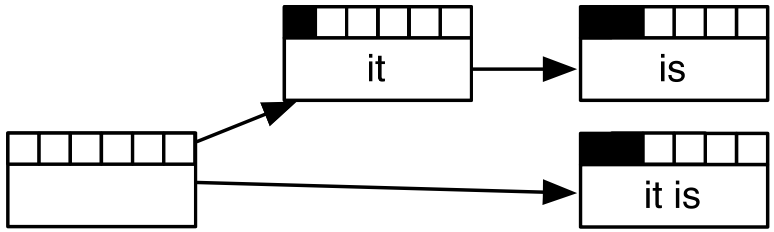




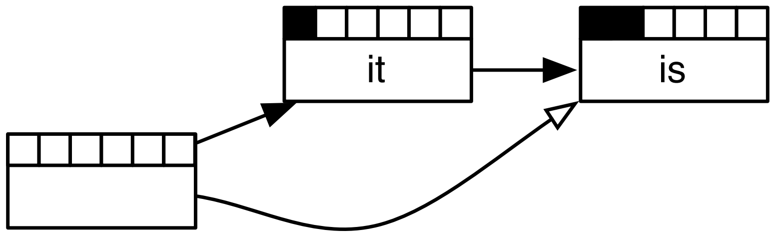

backtrack from highest scoring complete hypothesis

# Recombination

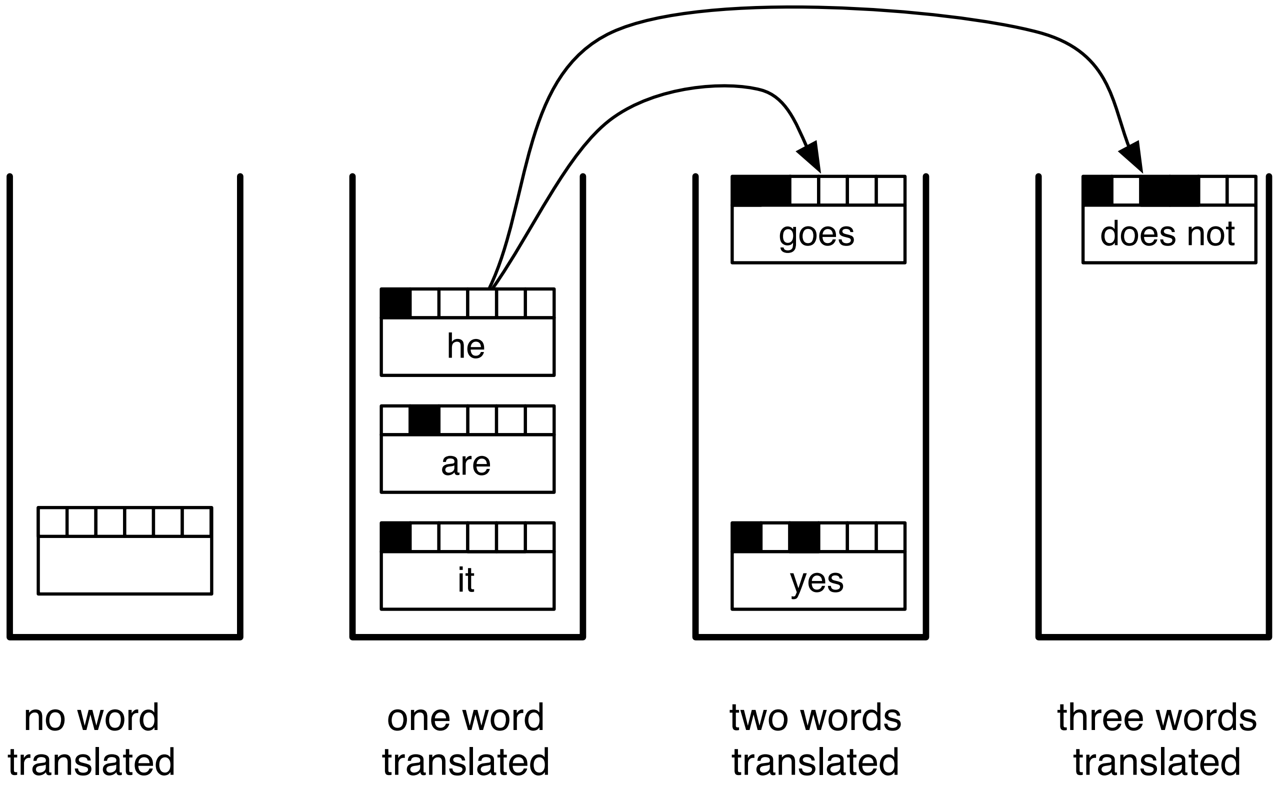
- Two hypothesis paths lead to two matching hypotheses
  - same number of foreign words translated
  - same English words in the output
  - different scores




- Worse hypothesis is dropped



# Stacks

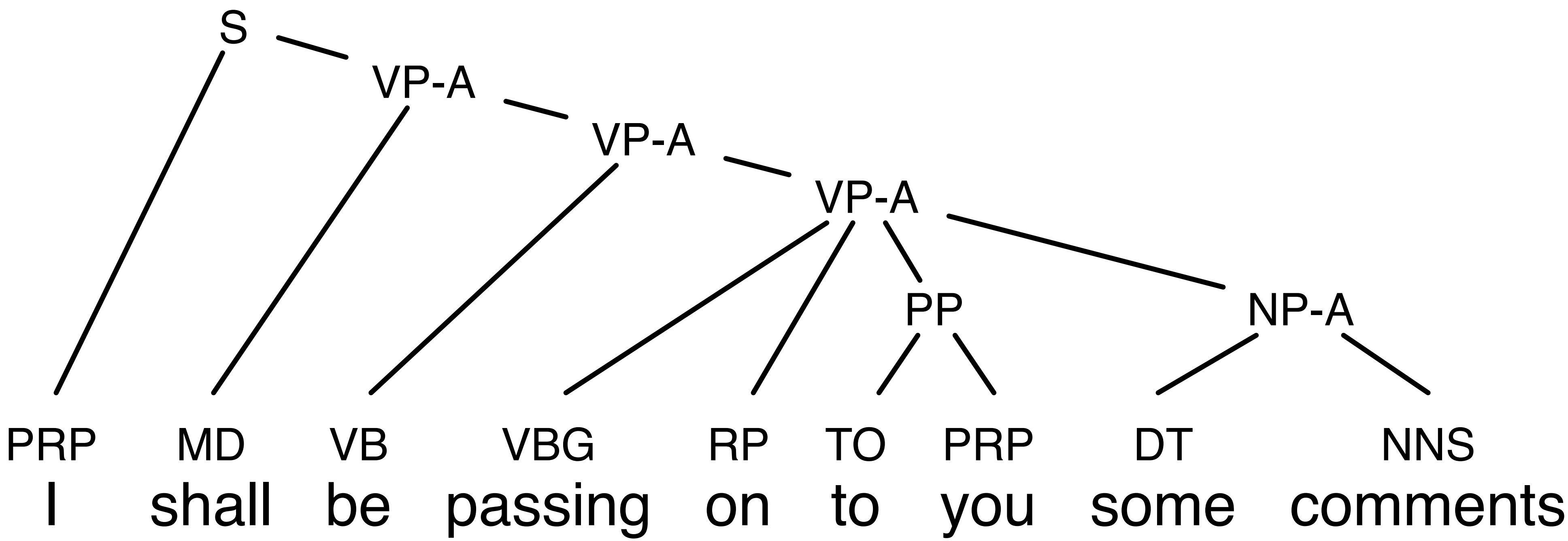



- Hypothesis expansion in a stack decoder
  - translation option is applied to hypothesis
  - new hypothesis is dropped into a stack further down

# syntax-based models

# Phrase Structure Grammar






Phrase structure grammar tree for an English sentence
(as produced Collins' parser)

# Synchronous Phrase Structure Grammar



- English rule

$$\text{NP} \rightarrow \text{DET JJ NN}$$

- French rule

$$\text{NP} \rightarrow \text{DET NN JJ}$$

- Synchronous rule (indices indicate alignment):

$$\text{NP} \rightarrow \text{DET}_1\ \text{NN}_2\ \text{JJ}_3 \mid \text{DET}_1\ \text{JJ}_3\ \text{NN}_2$$

# Synchronous Grammar Rules



- Nonterminal rules

$$\text{NP} \rightarrow \text{DET}_1\ \text{NN}_2\ \text{JJ}_3 \mid \text{DET}_1\ \text{JJ}_3\ \text{NN}_2$$

- Terminal rules

$$\text{N} \rightarrow \text{maison} \mid \text{house}$$

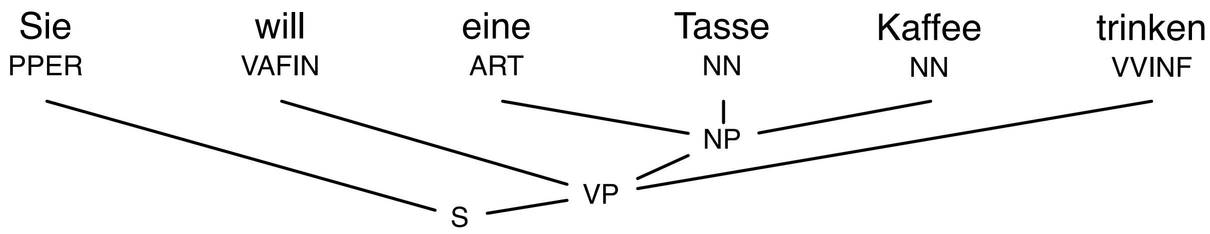
$$\text{NP} \rightarrow \text{la maison bleue} \mid \text{the blue house}$$

- Mixed rules

$$\text{NP} \rightarrow \text{la maison JJ}_1 \mid \text{the JJ}_1\ \text{house}$$

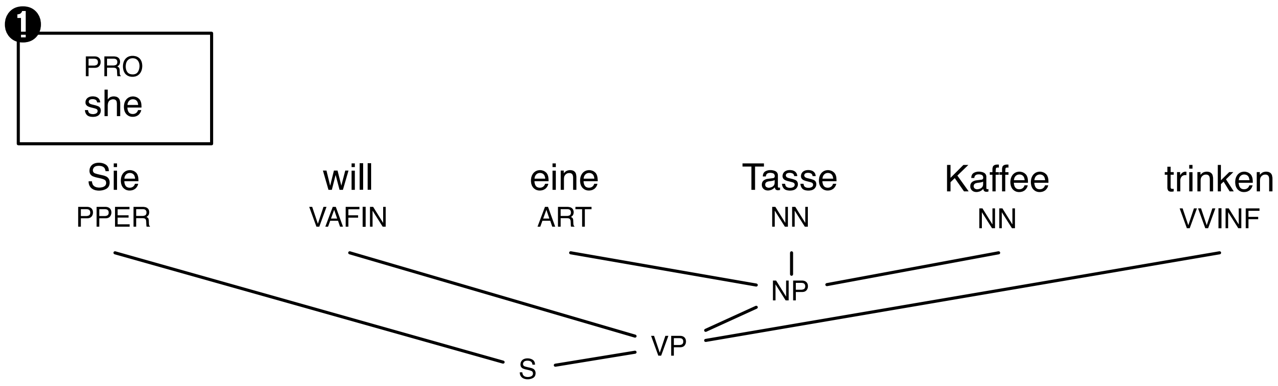
# Syntax Decoding





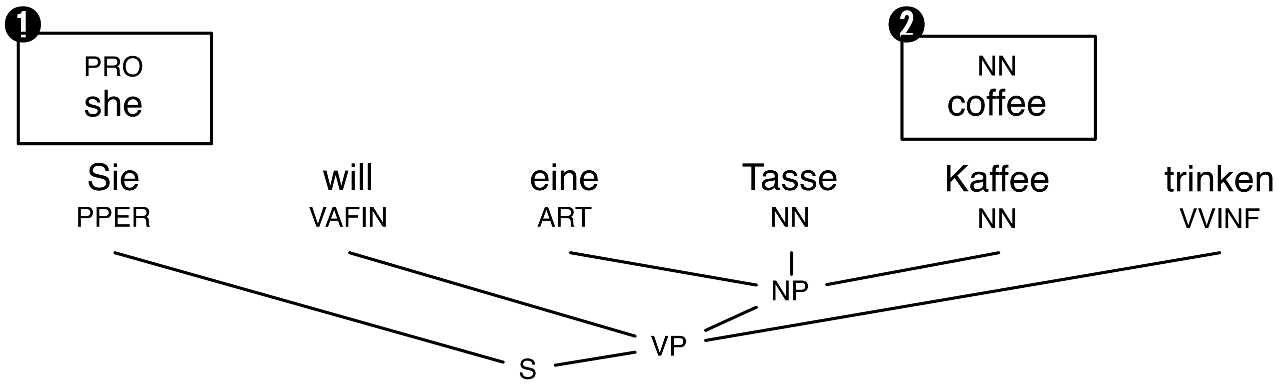

German input sentence with tree

# Syntax Decoding



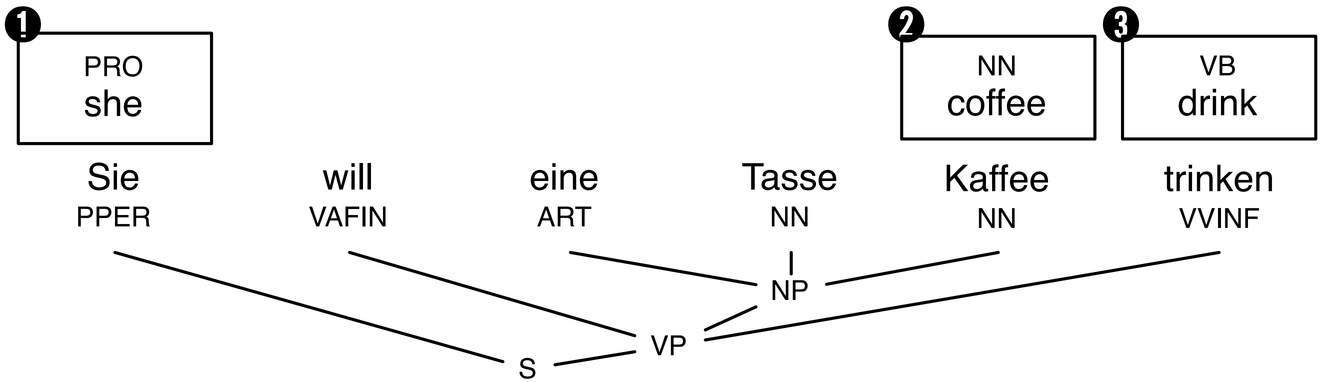

Purely lexical rule: filling a span with a translation (a constituent in the chart)

# Syntax Decoding




Purely lexical rule: filling a span with a translation (a constituent in the chart)

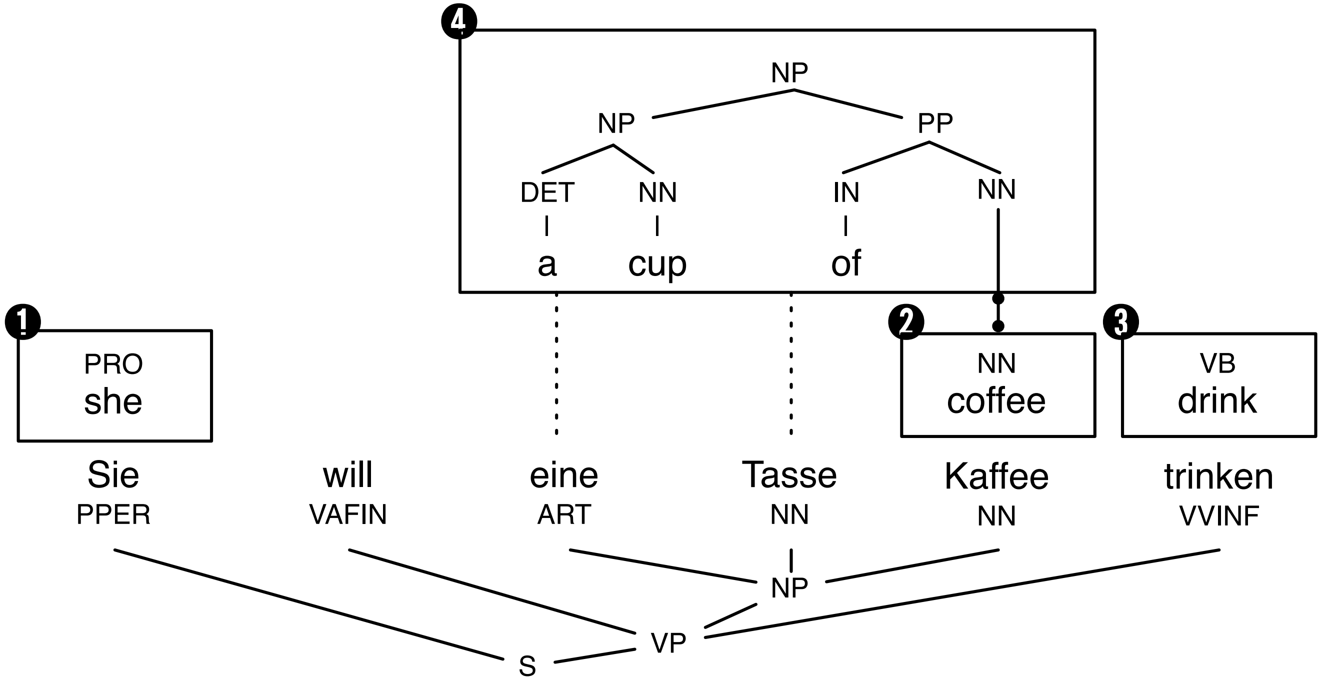
# Syntax Decoding




Purely lexical rule: filling a span with a translation (a constituent in the chart)

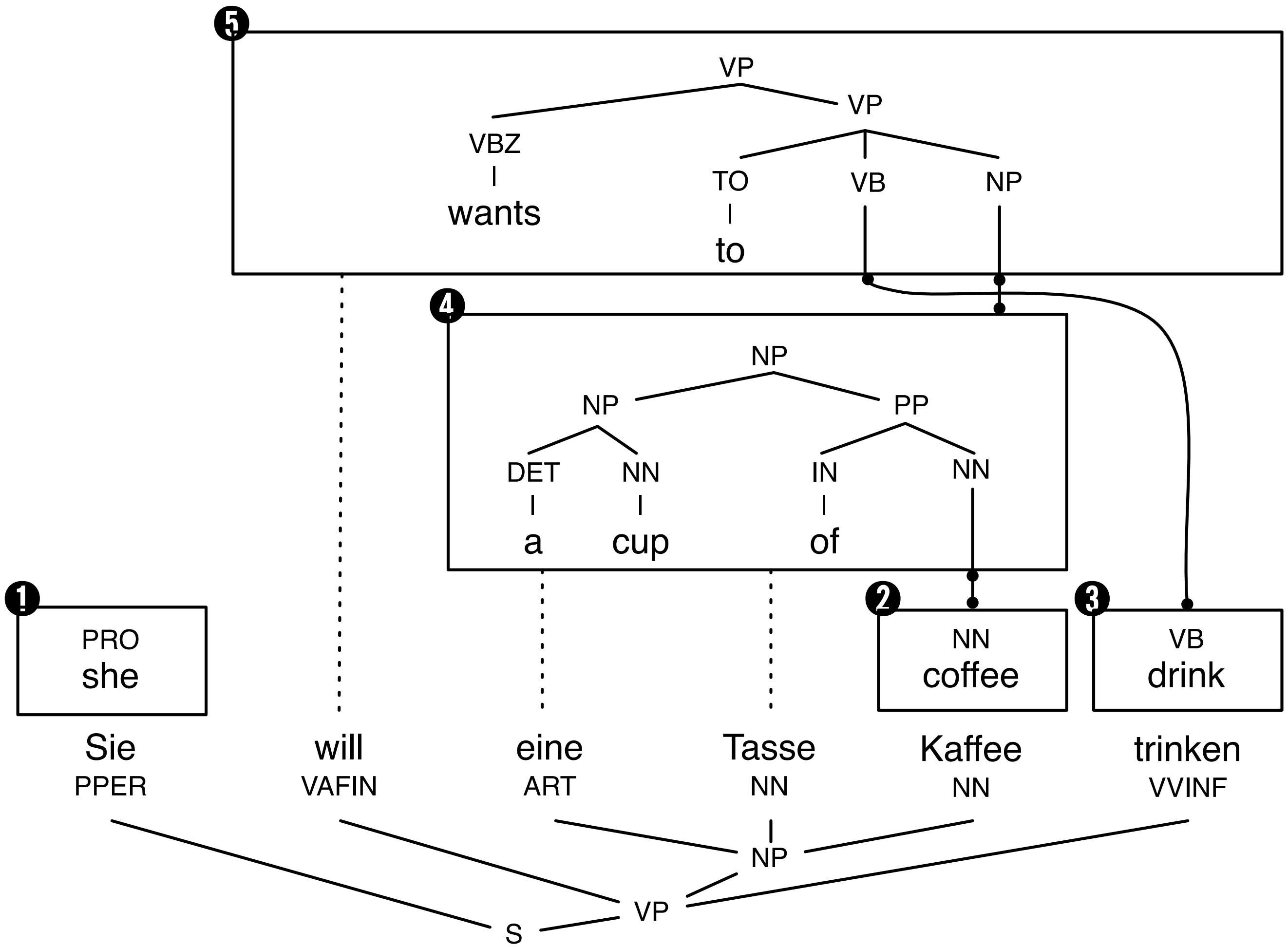
# Syntax Decoding






Complex rule: matching underlying constituent spans, and covering words

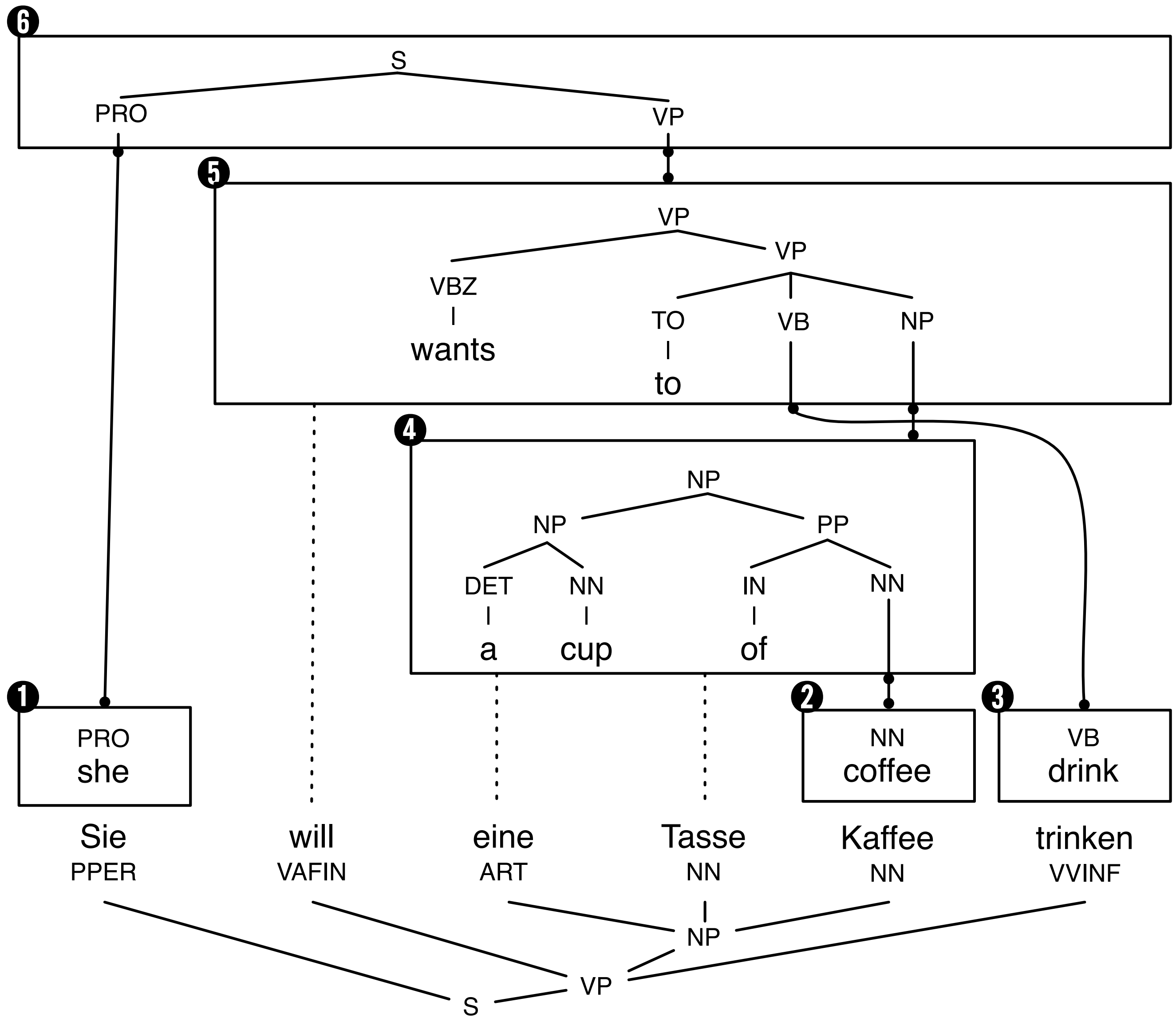
# Syntax Decoding




Complex rule with reordering

# Syntax Decoding





# neural language models

# N-Gram Backoff Language Model



- Previously, we approximated

$$p(W) = p(w_1, w_2, ..., w_n)$$

- ... by applying the chain rule

$$p(W) = \sum_i p(w_i|w_1, ..., w_{i-1})$$

- ... and limiting the history (Markov order)

$$p(w_i|w_1, ..., w_{i-1}) \simeq p(w_i|w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$$

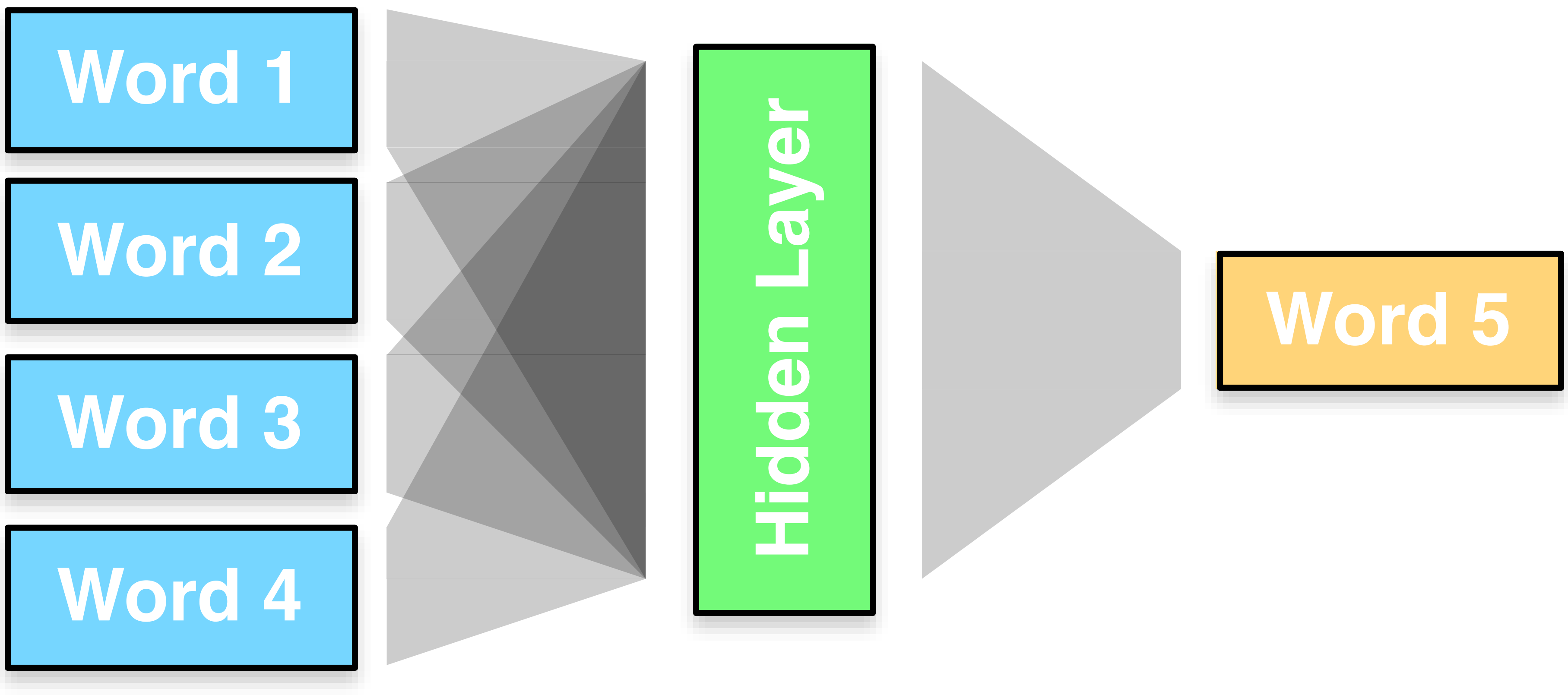
- Each $p(w_i|w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$ may not have enough statistics to estimate

→ we back off to $p(w_i|w_{i-3}, w_{i-2}, w_{i-1})$, $p(w_i|w_{i-2}, w_{i-1})$, etc., all the way to $p(w_i)$

– exact details of backing off get complicated — "interpolated Kneser-Ney"

# First Sketch





# Representing Words

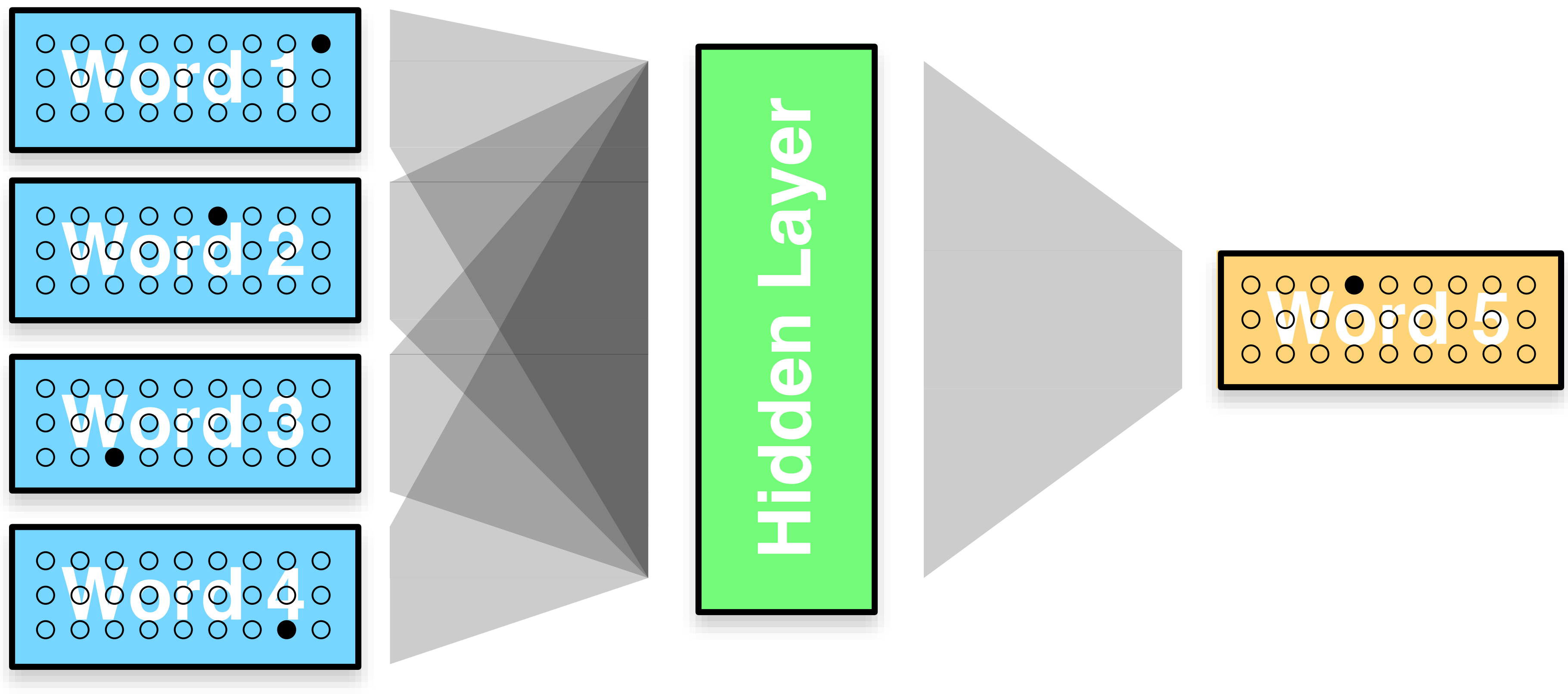


- Words are represented with a one-hot vector, e.g.,
  - dog = (0,0,0,0,1,0,0,0,0,....)
  - cat = (0,0,0,0,0,0,0,1,0,....)
  - eat = (0,1,0,0,0,0,0,0,0,....)

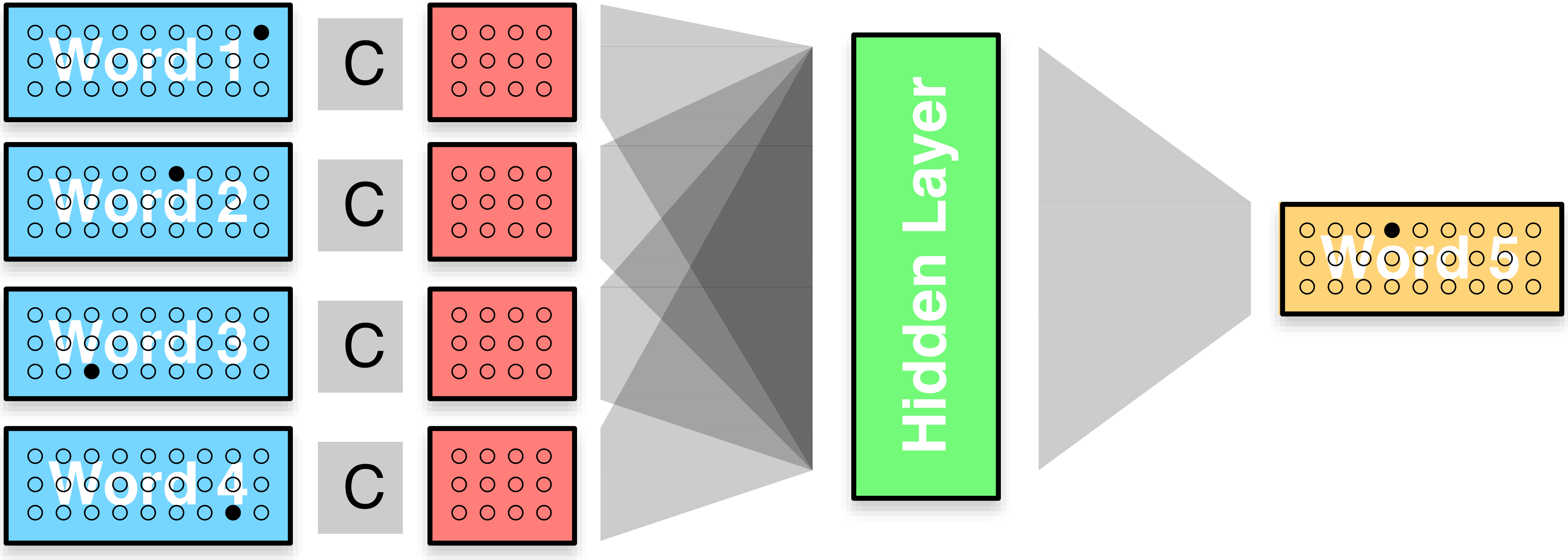
- That's a large vector!

# Second Sketch





# Add a Hidden Layer




- Map each word first into a lower-dimensional real-valued space
- Shared weight matrix $C$

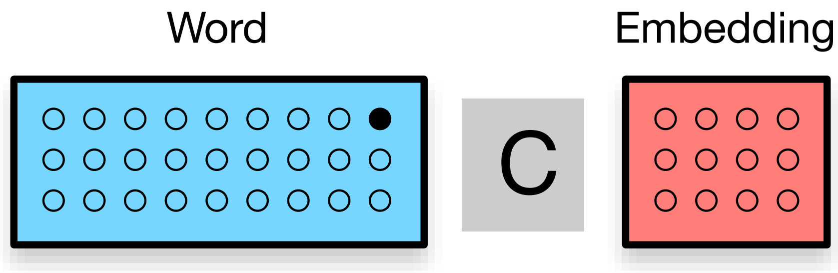
# Details (Bengio et al., 2003)



- Add direct connections from embedding layer to output layer
- Activation functions
  - input→embedding: none
  - embedding→hidden: tanh
  - hidden→output: softmax
- Training
  - loop through the entire corpus
  - update between predicted probabilities and 1-hot vector for output word

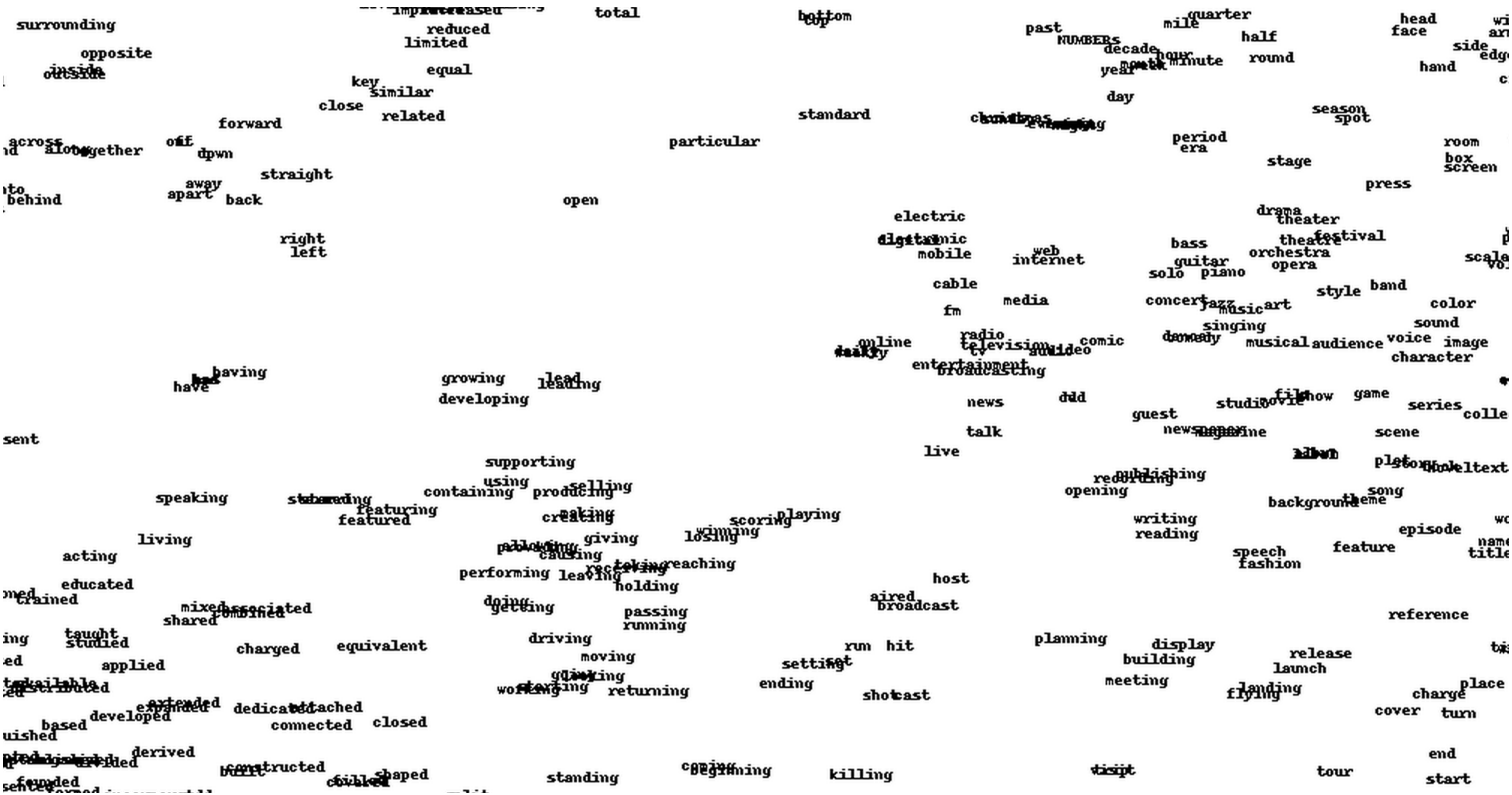
# Word Embeddings



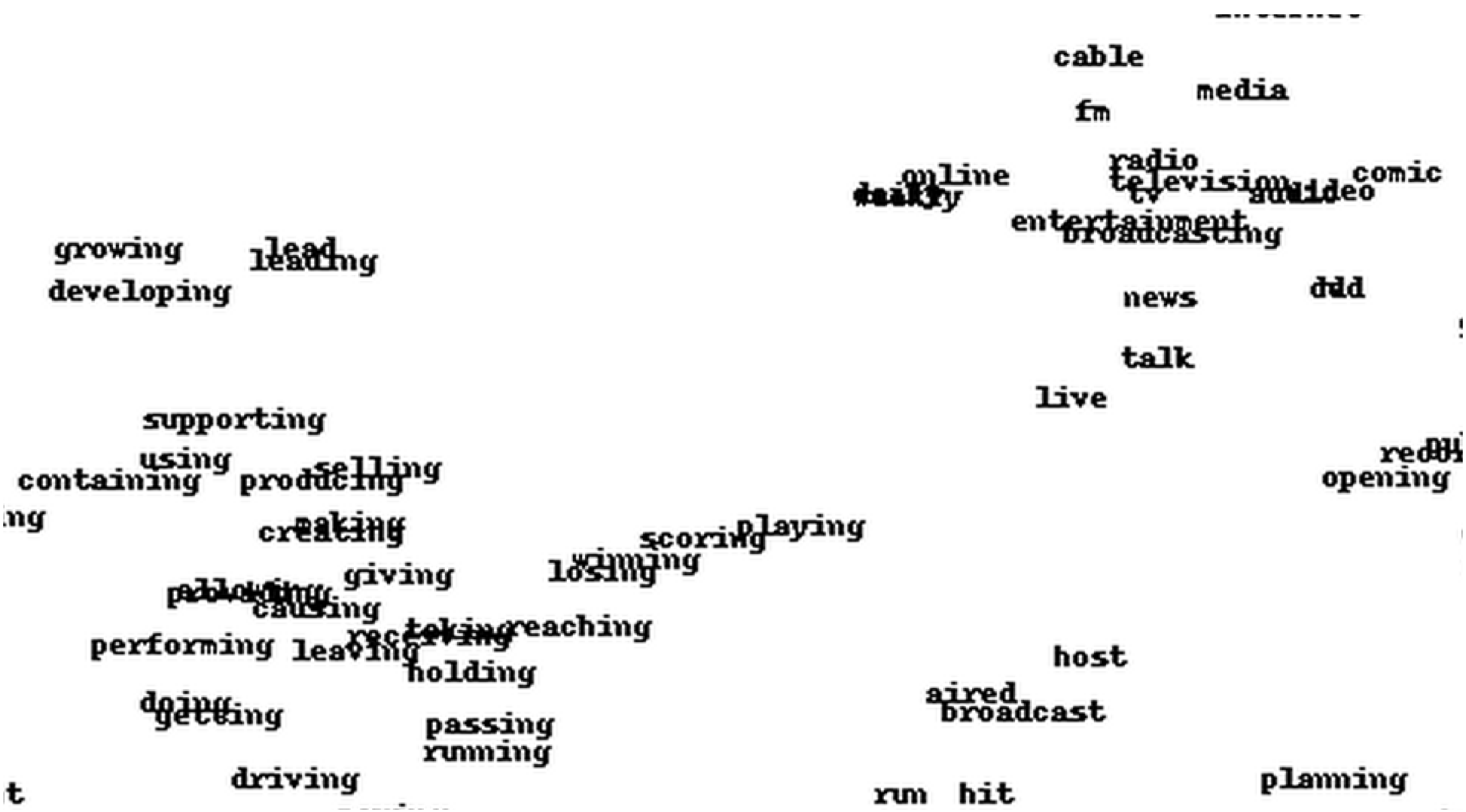



- By-product: embedding of word into continuous space
- Similar contexts → similar embedding
- Recall: distributional semantics

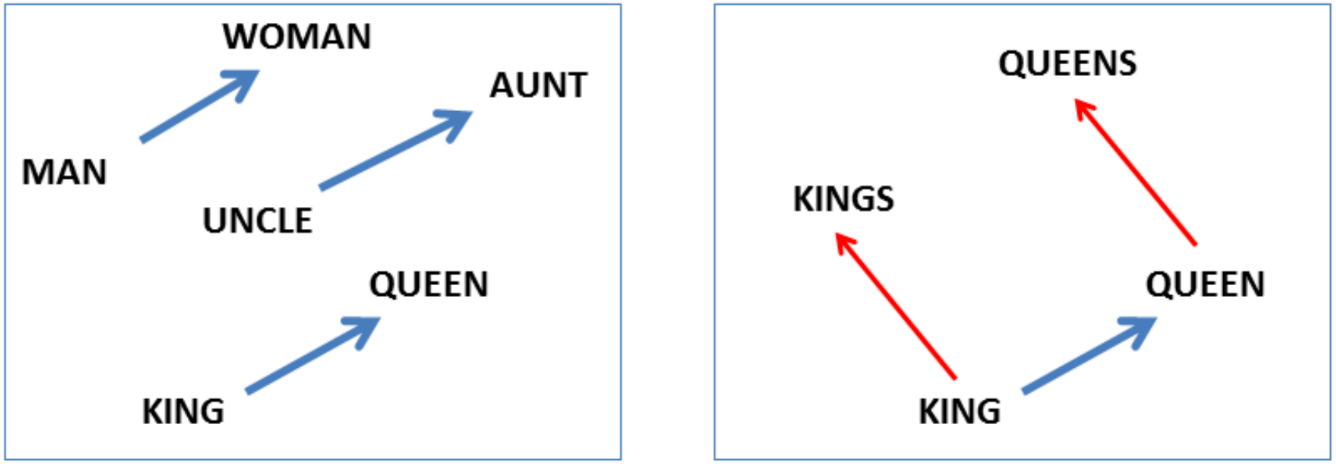
# Word Embeddings



# Word Embeddings



# Are Word Embeddings Magic?



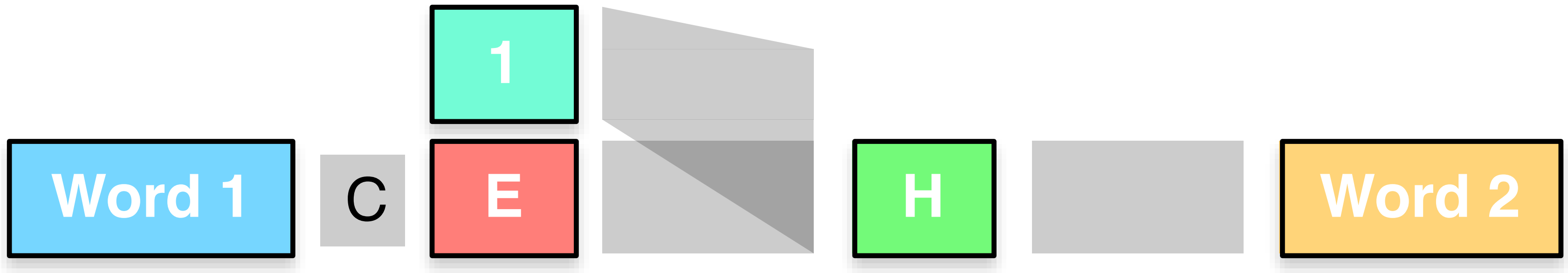

- Morphosyntactic regularities (Mikolov et al., 2013)
  - adjectives base form vs. comparative, e.g., good, better
  - nouns singular vs. plural, e.g., year, years
  - verbs present tense vs. past tense, e.g., see, saw

- Semantic regularities
  - clothing is to shirt as dish is to bowl
  - evaluated on human judgment data of semantic similarities

# recurrent neural networks

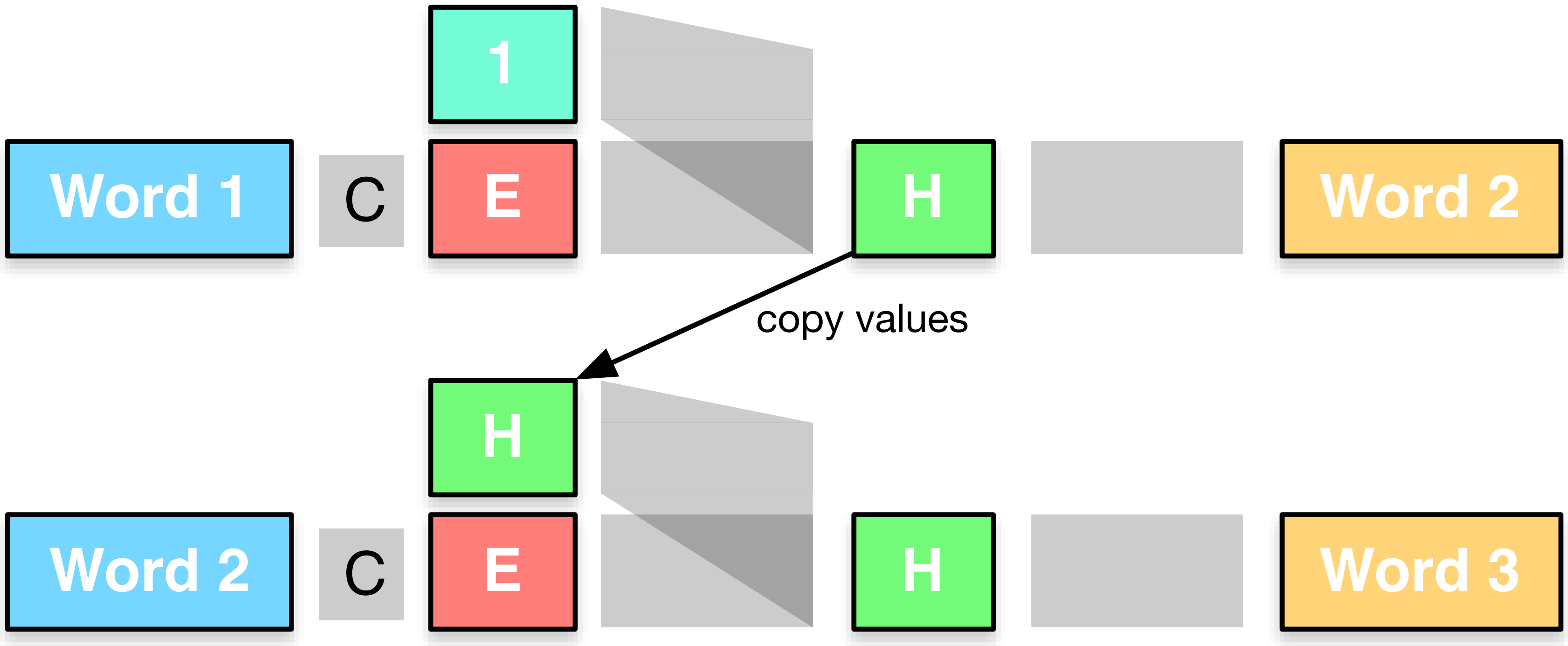
# Recurrent Neural Networks





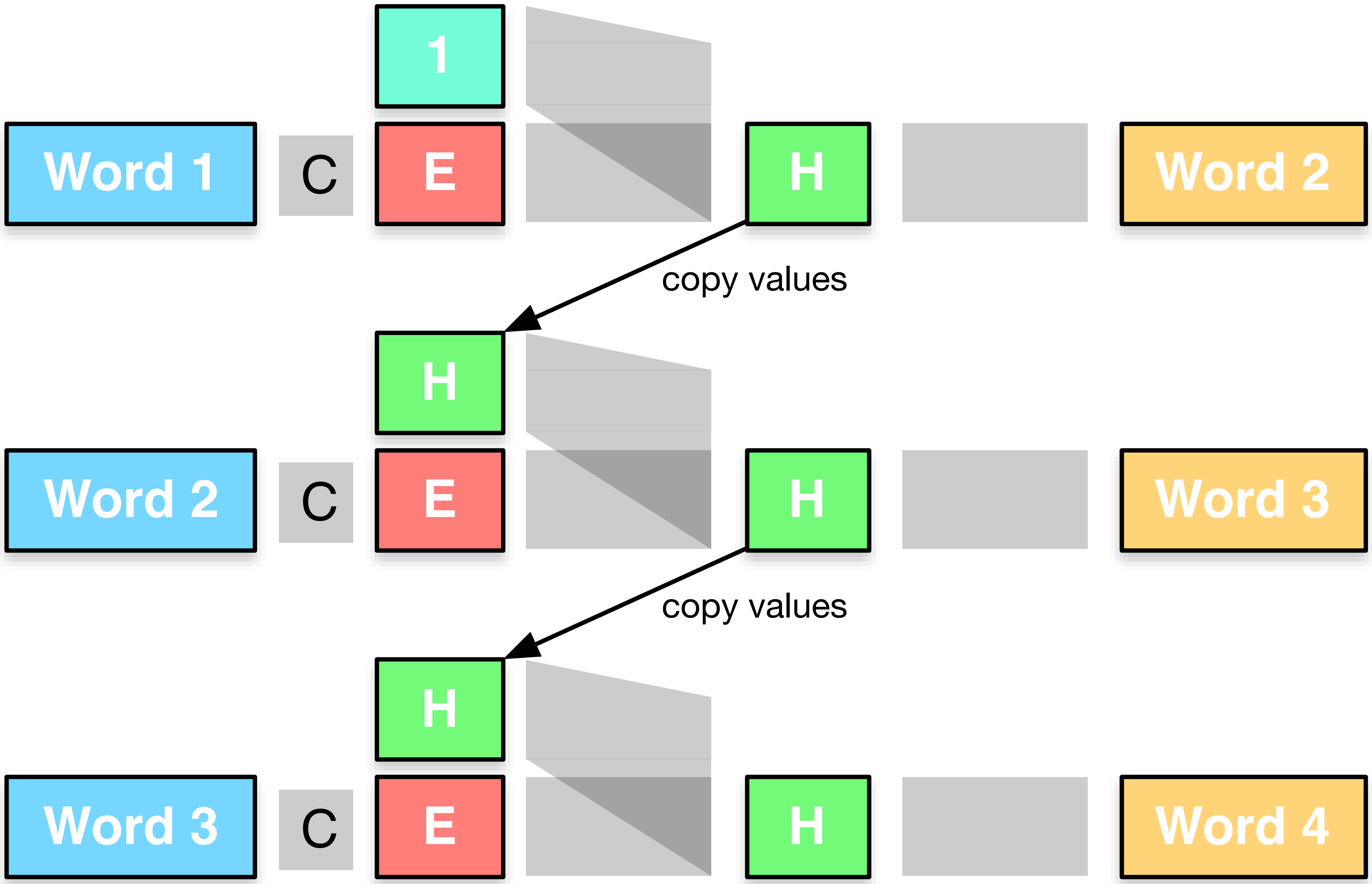

- Start: predict second word from first
- Mystery layer with nodes all with value 1

# Recurrent Neural Networks

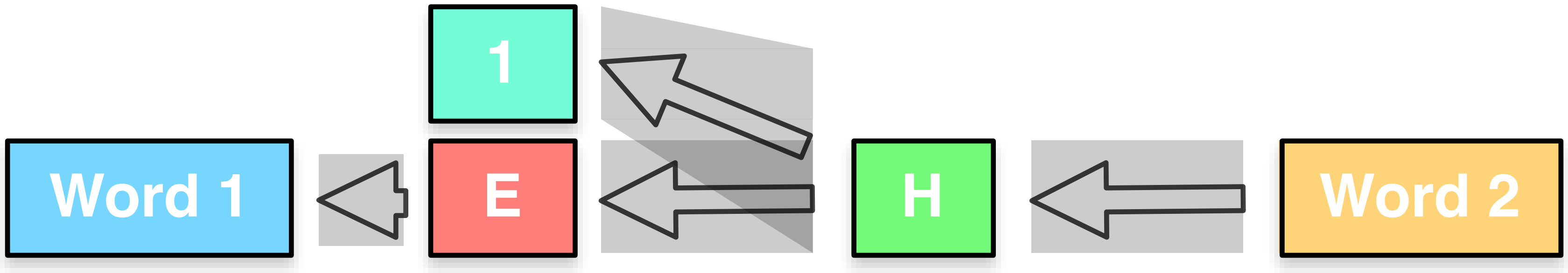


# Recurrent Neural Networks





# Training

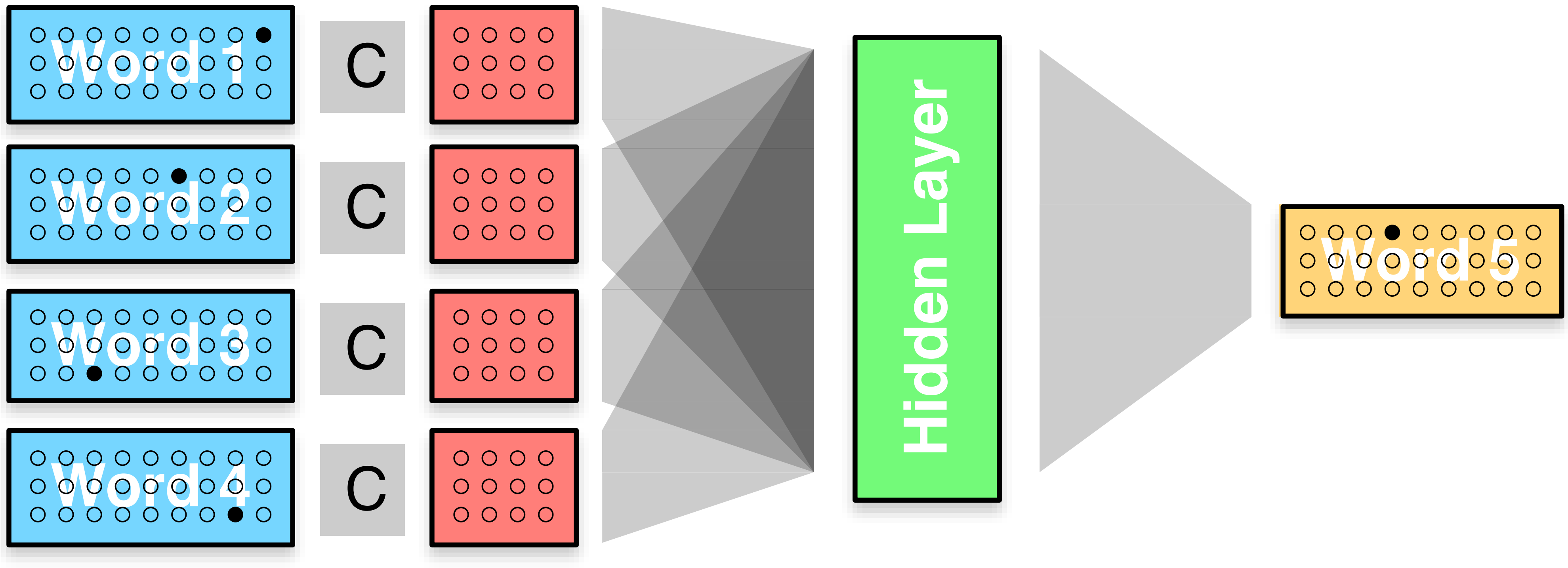



- Process first training example
- Update weights with back-propagation

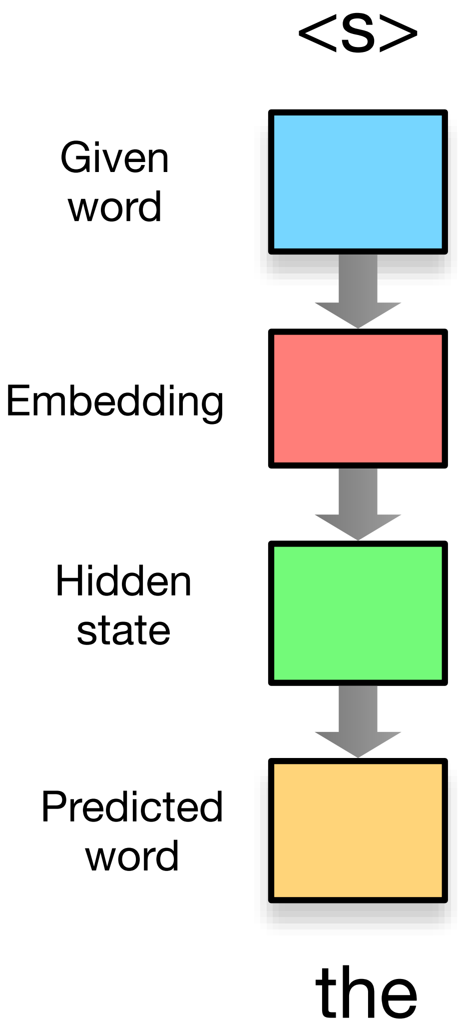
# neural translation model

# Feed Forward Neural Language Model





# Recurrent Neural Language Model

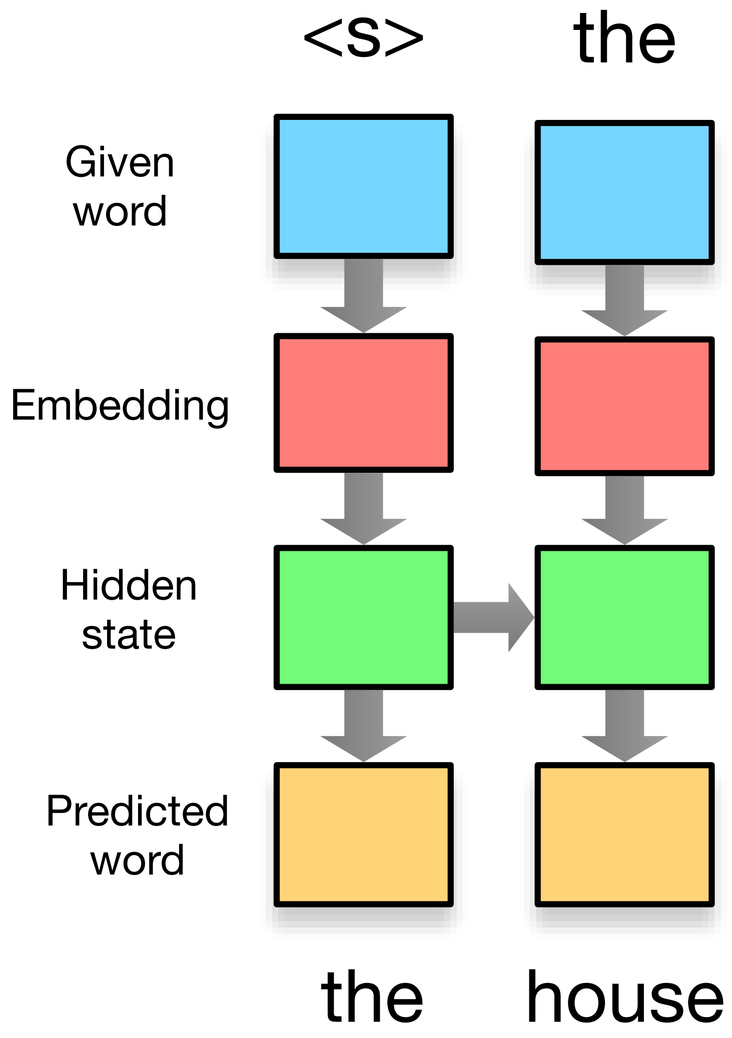



Predict
the first word
of a sentence

Same as before,
just drawn top-down

# Recurrent Neural Language Model



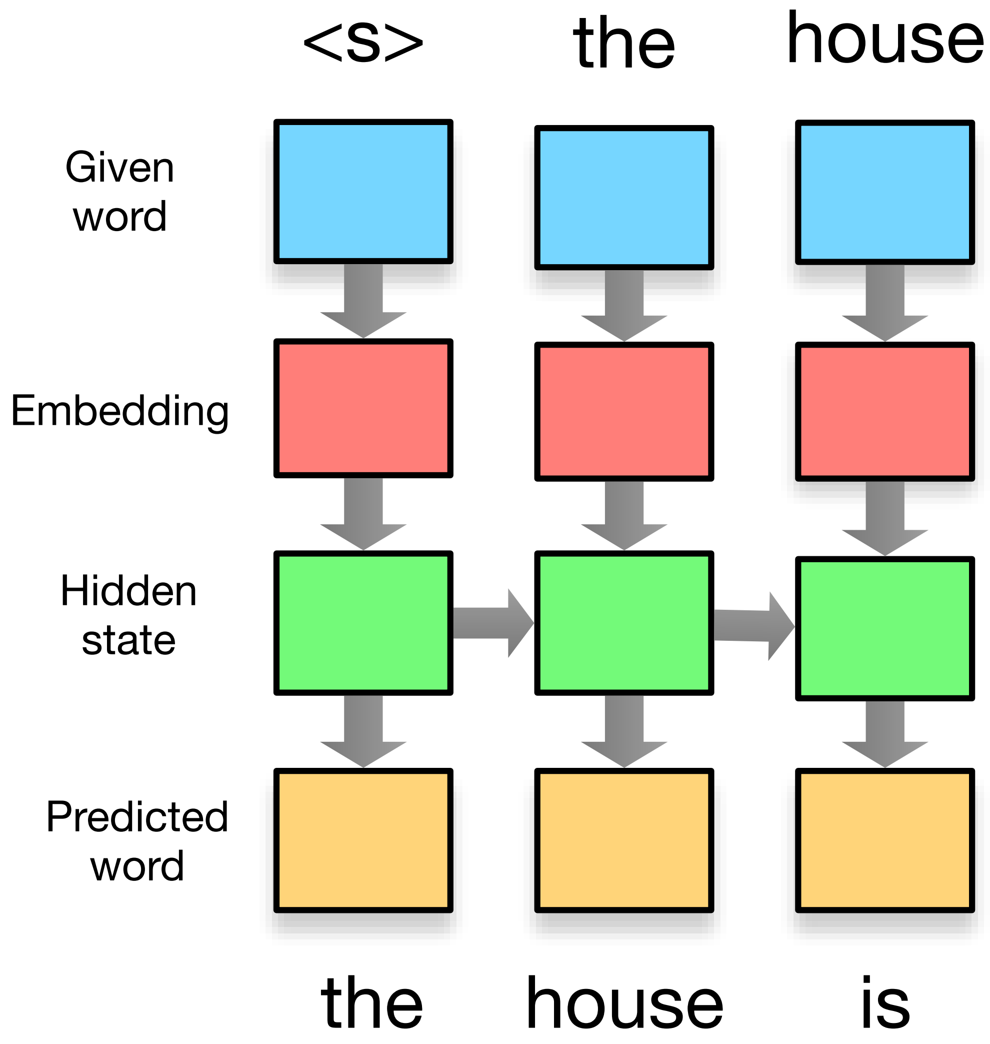



Predict
the second word
of a sentence

Re-use hidden state
from
first word prediction

# Recurrent Neural Language Model



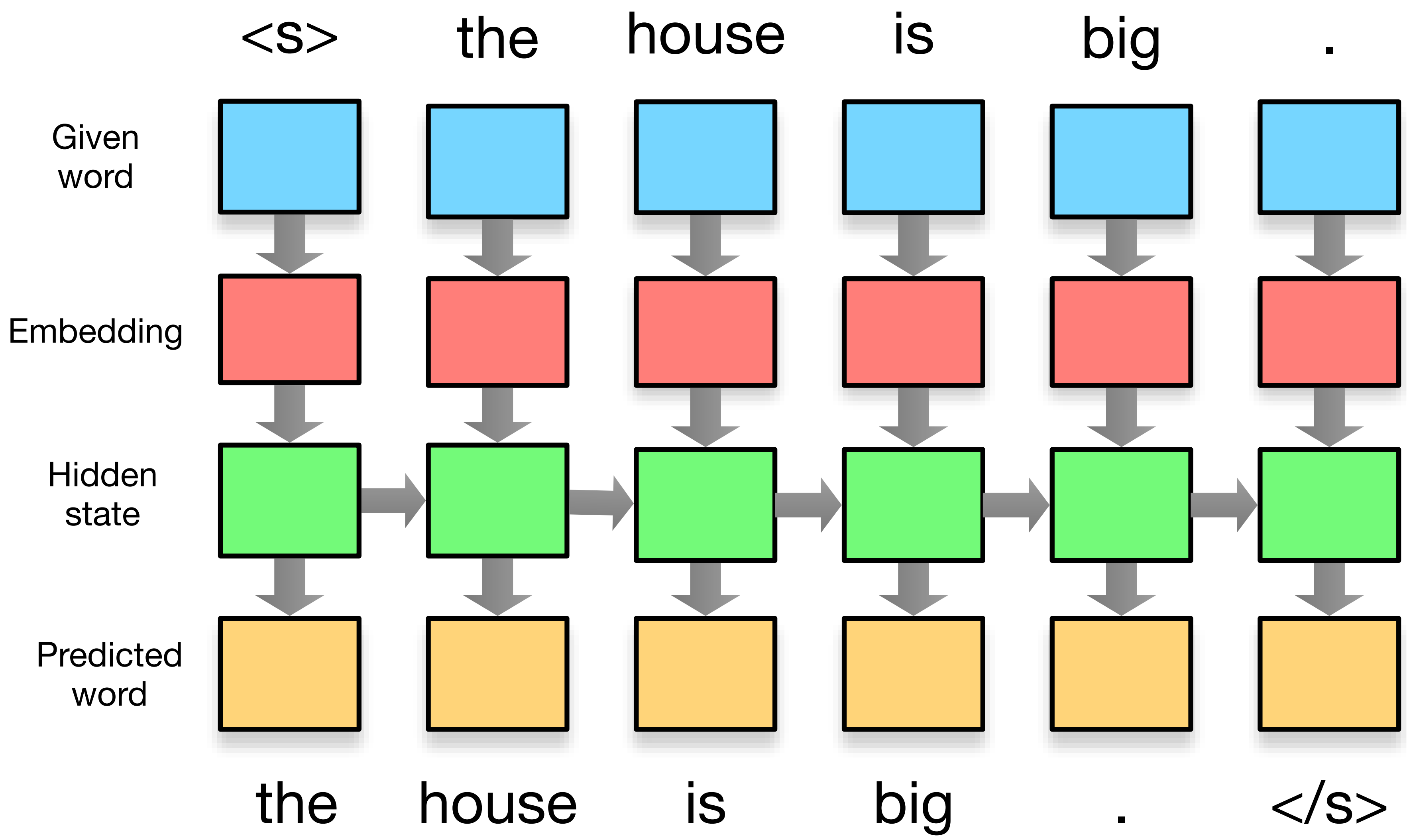



Predict
the third word
of a sentence

... and so on

# Recurrent Neural Language Model





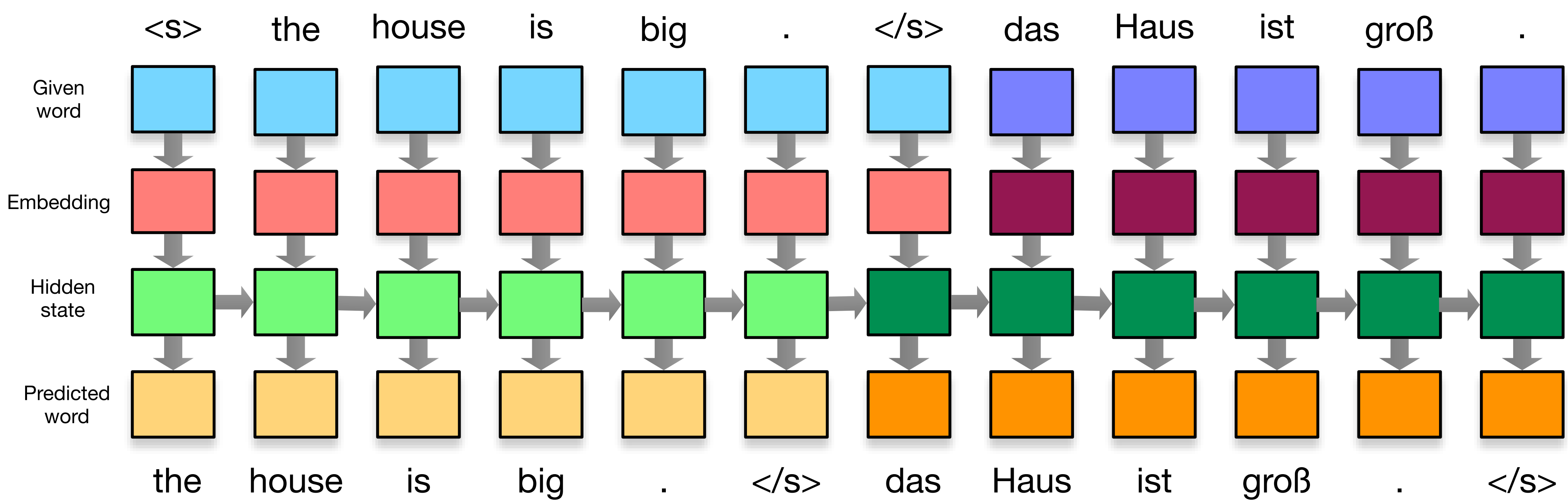
# Recurrent Neural Translation Model



- We predicted the words of a sentence

- Why not also predict their translations?

# Encoder-Decoder Model




- Obviously madness

- Proposed by Google (Sutskever et al. 2014)

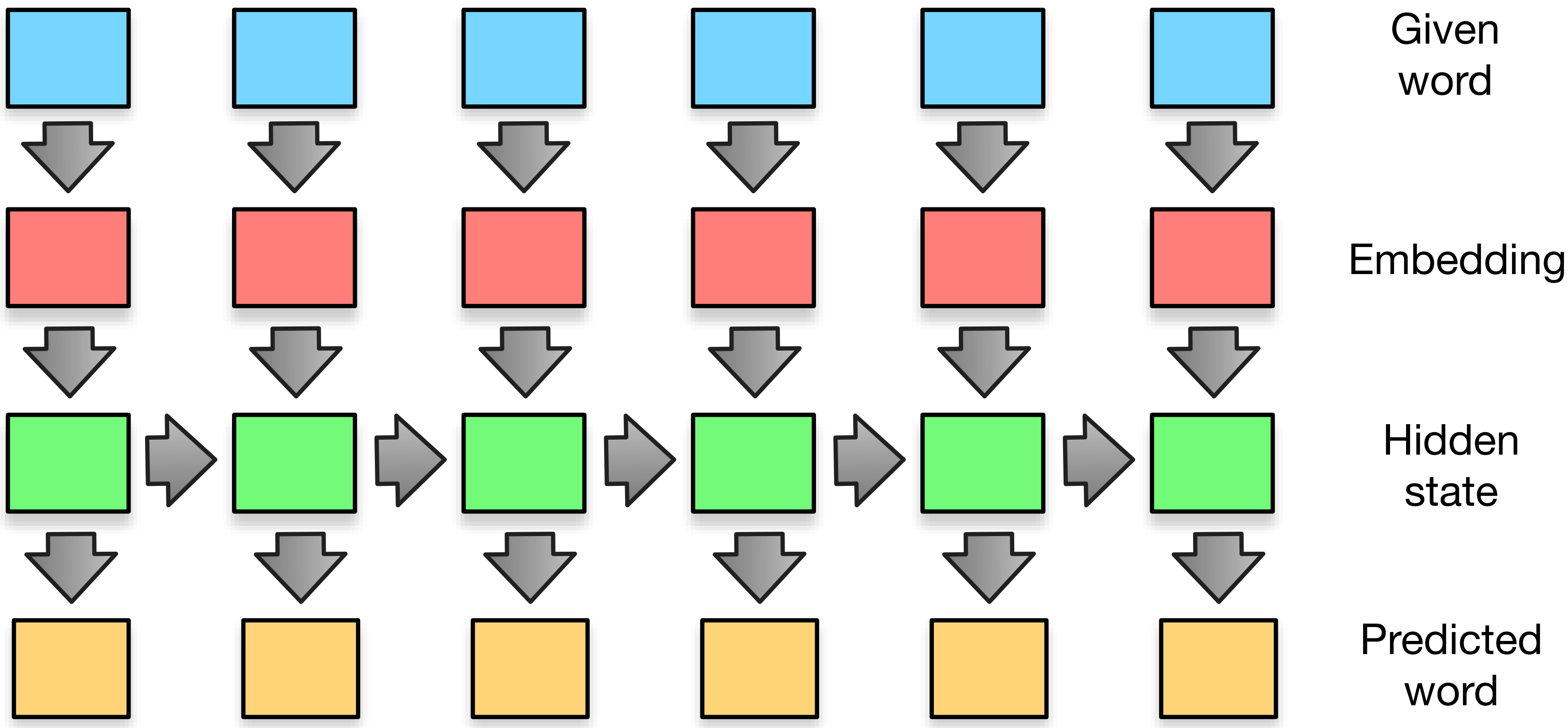
# What is Missing?

- Alignment of input words to output words

⇒ Solution: attention mechanism

# neural translation model with attention

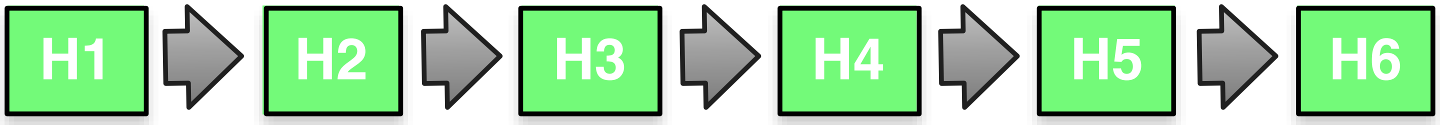
# Input Encoding




- Inspiration: recurrent neural network language model on the input side

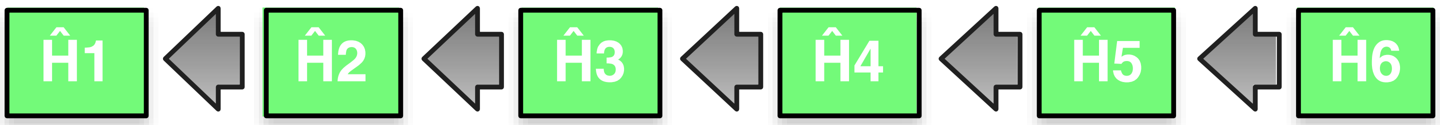
# Hidden Language Model States

- This gives us the hidden states



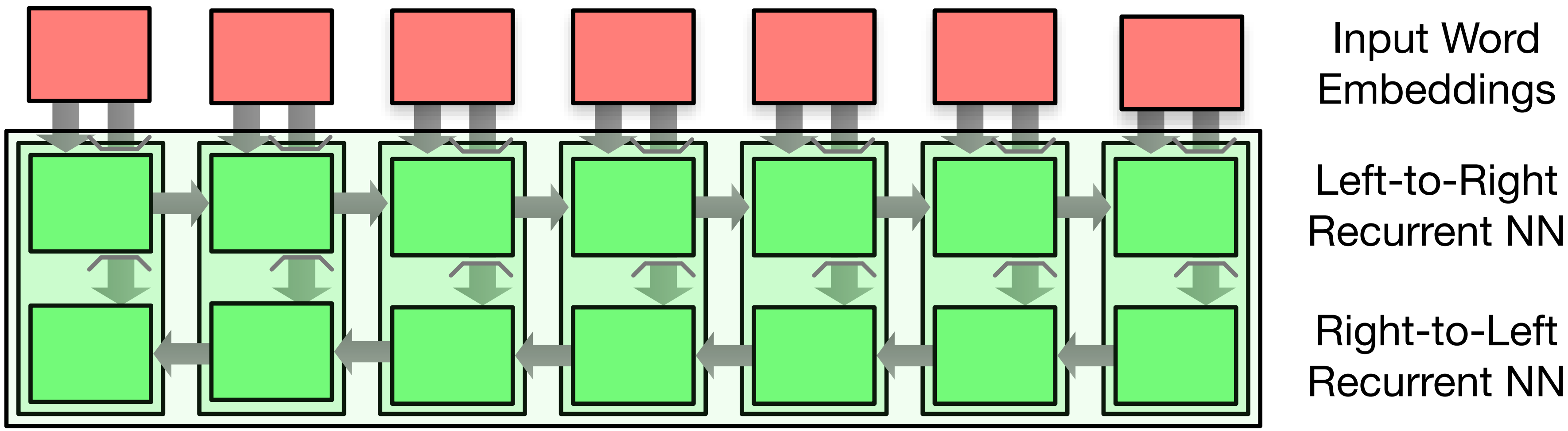

- These encode left context for each word

- Same process in reverse: right context for each word



# Input Encoder



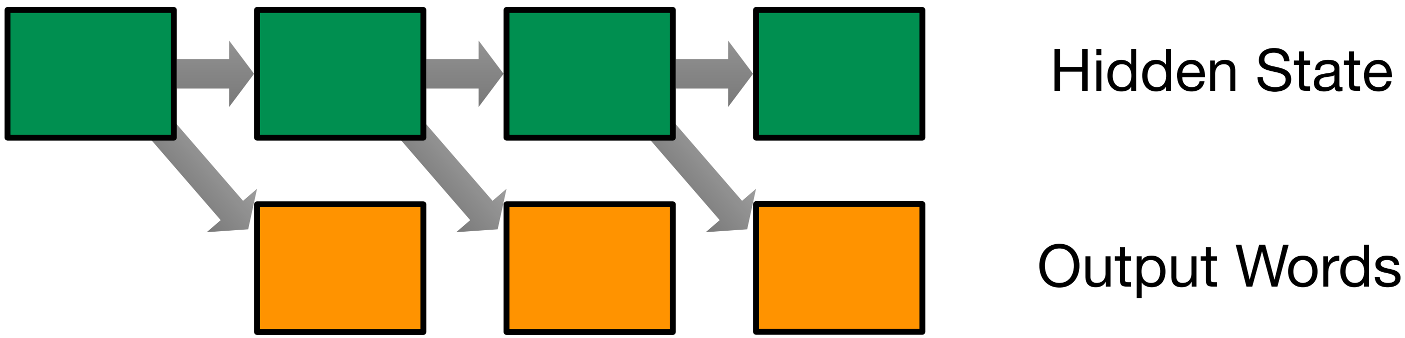



- Input encoder: concatenate bidirectional RNN states
- Each word representation includes full left and right sentence context

# Decoder



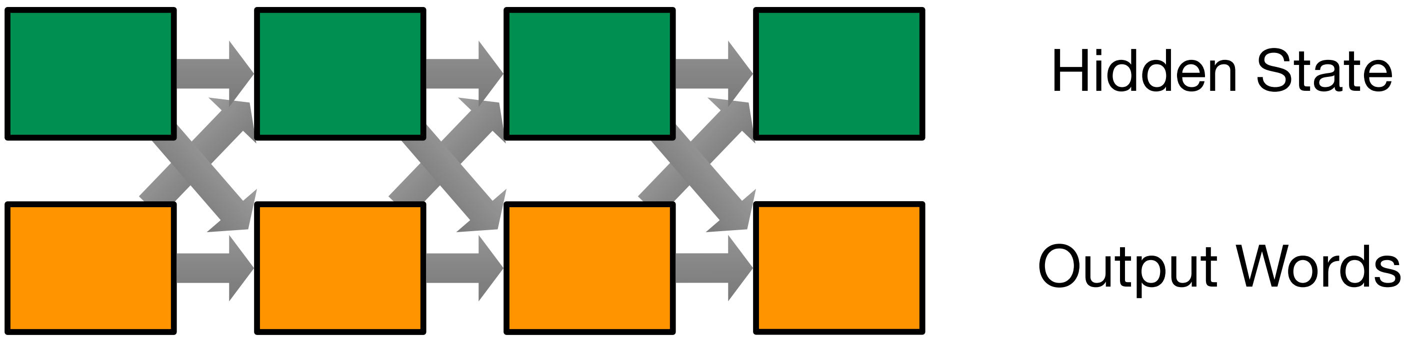
- We want to have a recurrent neural network predicting output words



# Decoder



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

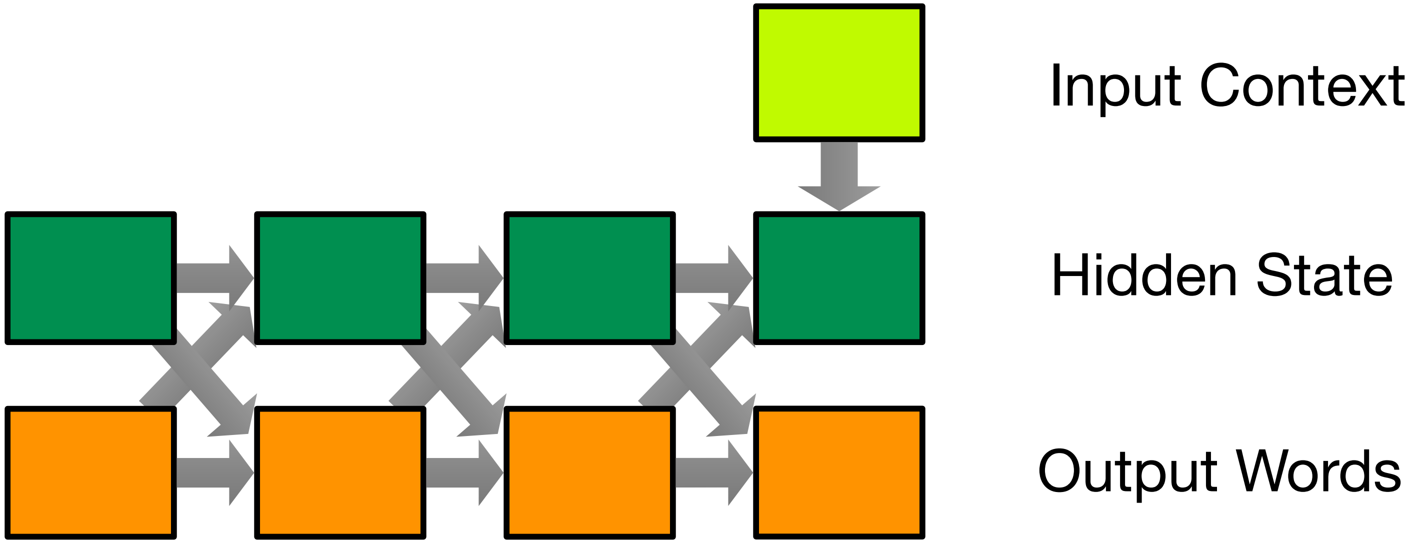



- We feed decisions on output words back into the decoder state

# Decoder



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

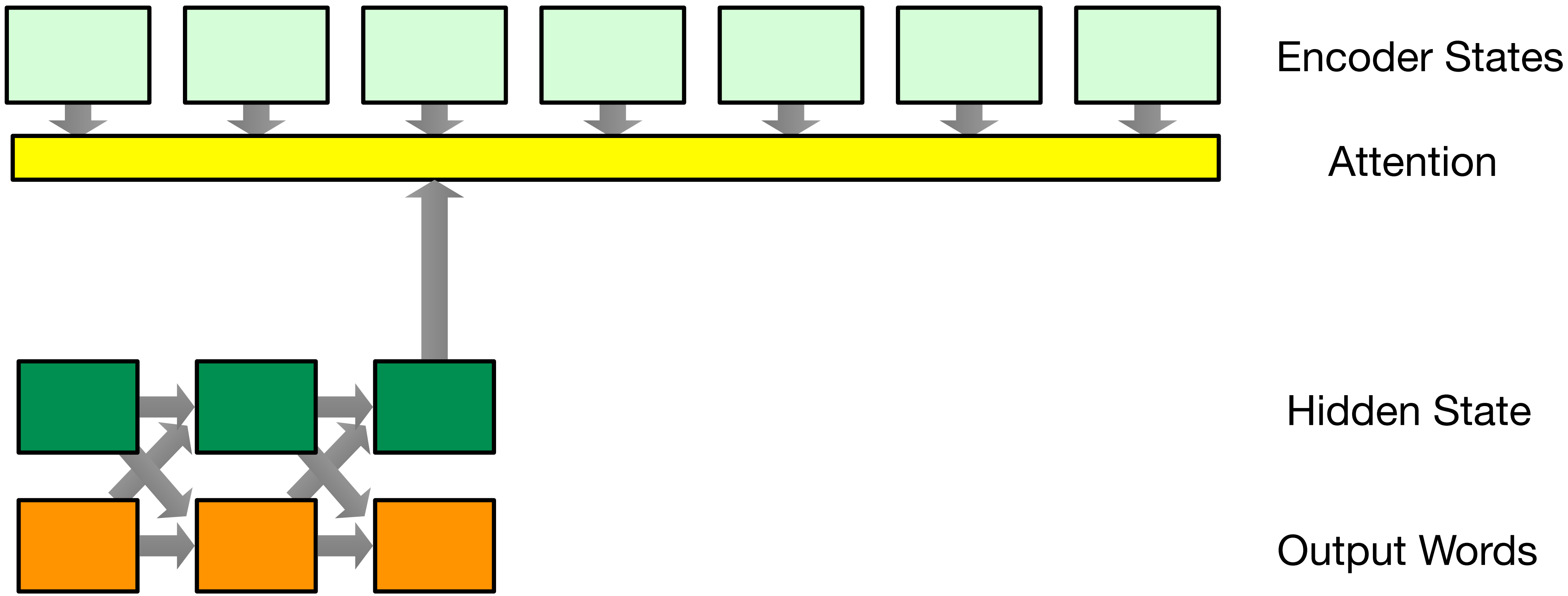



- We feed decisions on output words back into the decoder state

- Decoder state is also informed by the input context

# Attention

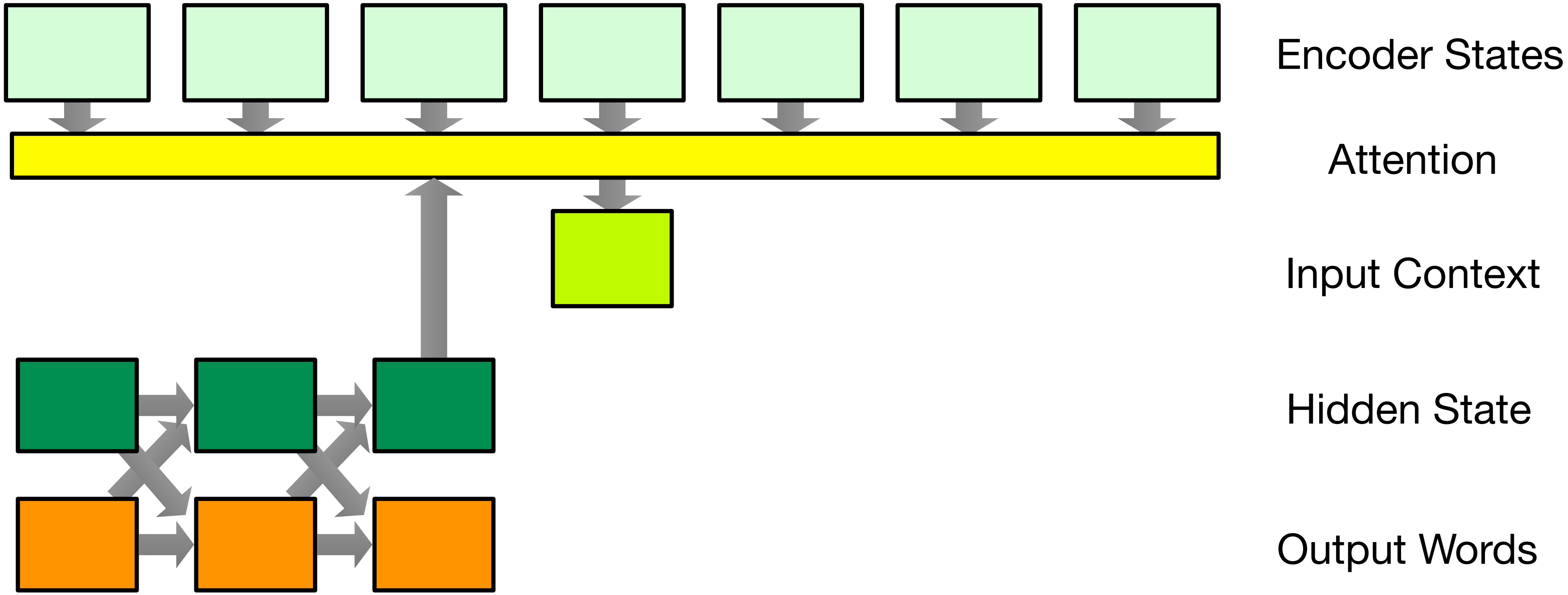





- Given what we have generated so far (decoder hidden state)
- ... which words in the input should we pay attention to (encoder states)?

# Attention



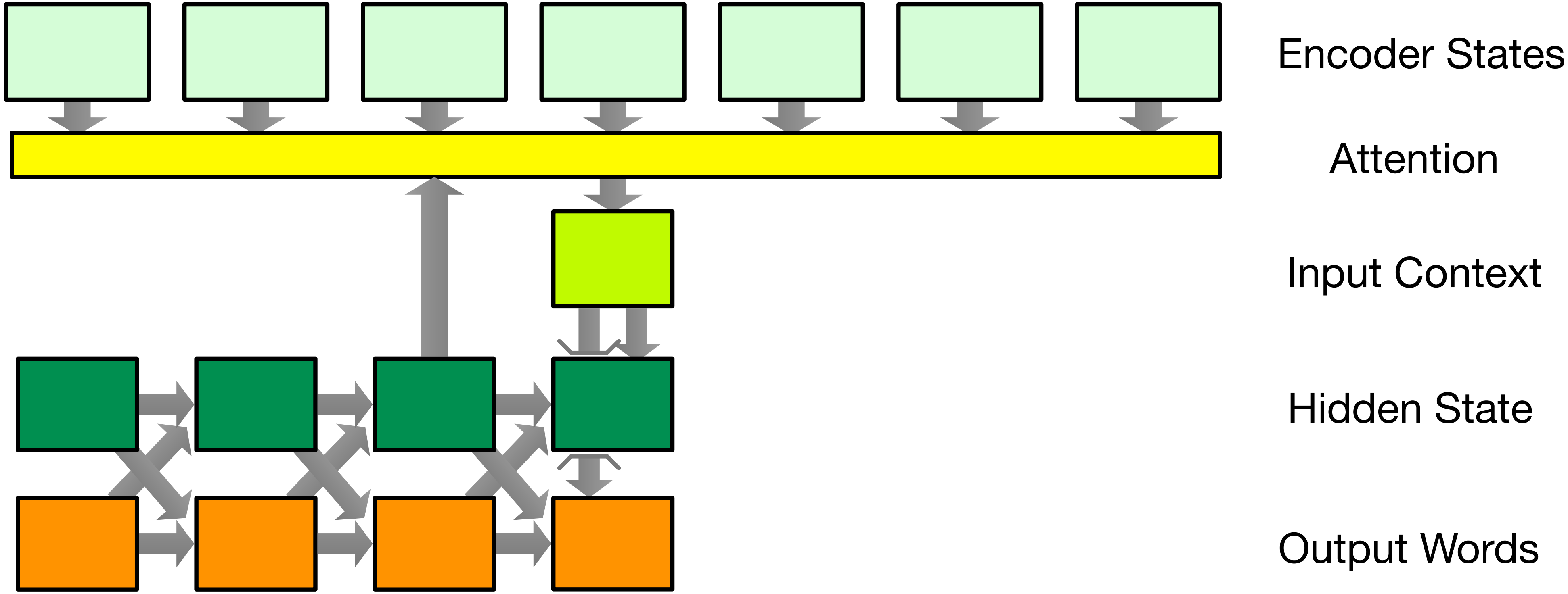



- Normalize attention (softmax)

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

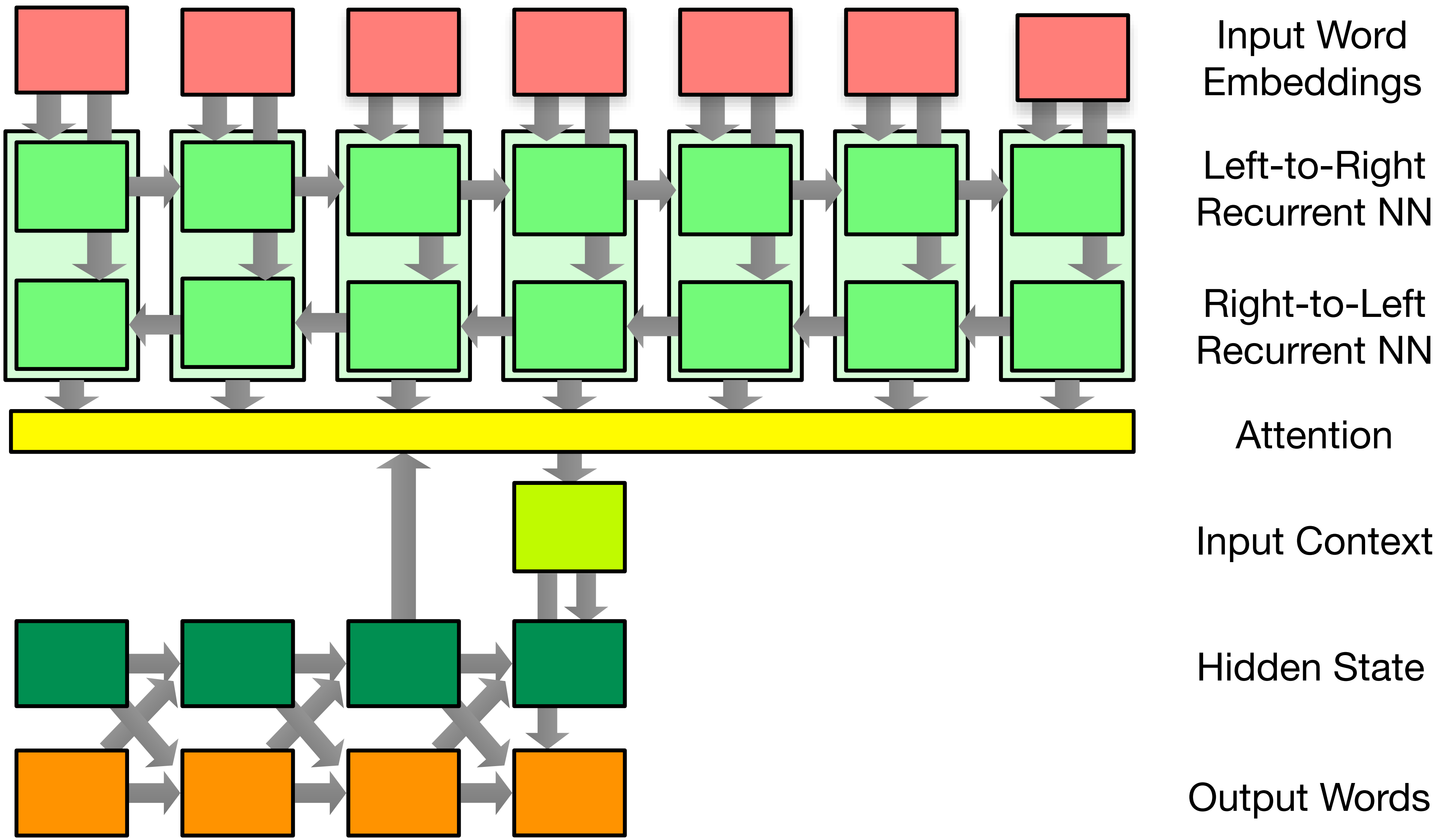
- Relevant input context: weigh input words according to attention: $c_i = \sum_j \alpha_{ij} h_j$

# Attention




- Use context to predict next hidden state and output word

# Encoder-Decoder with Attention



# questions?