

COMP90042 LECTURE 21

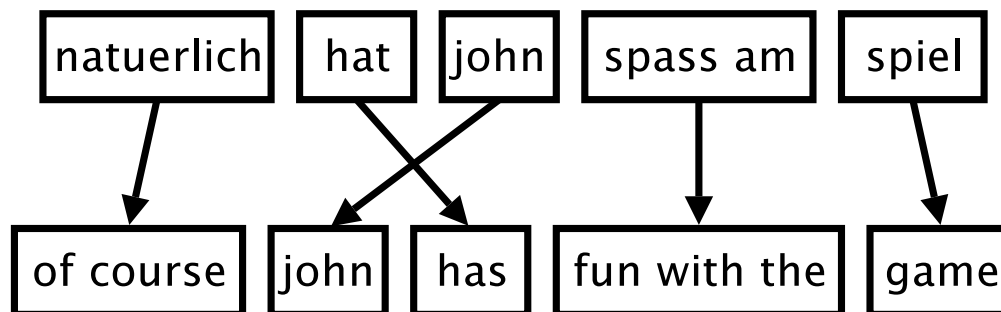
MT: PHRASE BASED & NEURAL ENCODER-DECODER

OVERVIEW

- ▶ Phrase based SMT
 - ▶ Scoring formula
 - ▶ Decoding algorithm
- ▶ Neural network ‘encoder-decoder’

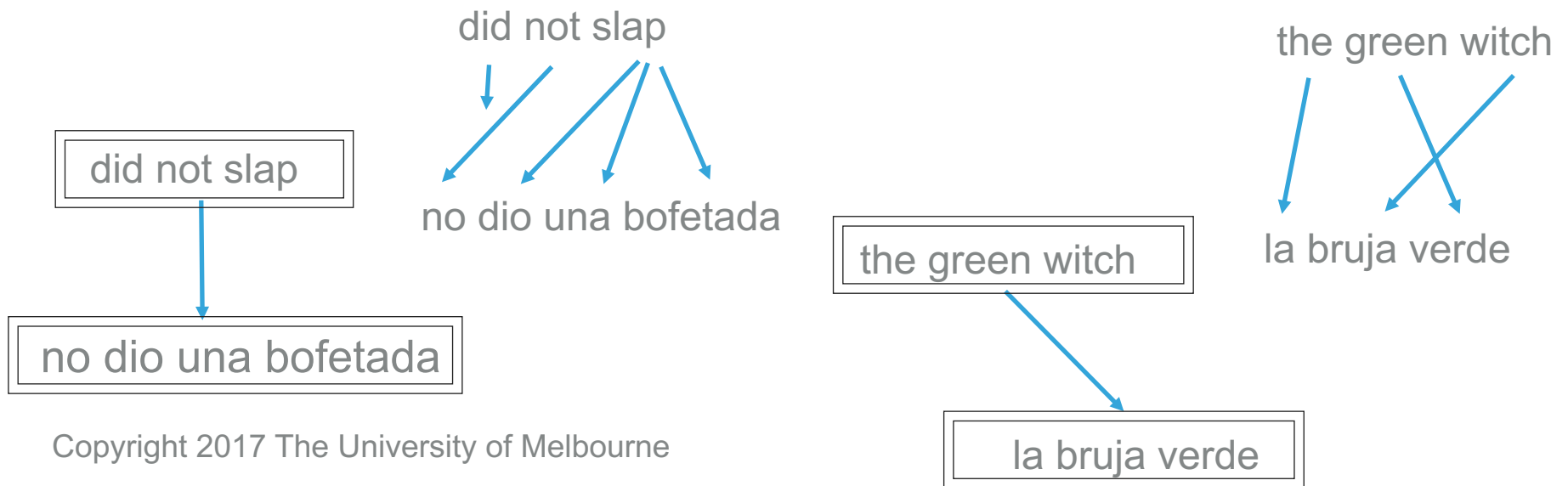
WORD- AND PHRASE-BASED MT

- ▶ Seen word based models of translation
 - ▶ now used for *alignment*, but not actual *translation*
 - ▶ overly simplistic formulation
- ▶ Phrase based MT
 - ▶ treats n-grams as translation units, referred to as ‘phrases’ (not linguistic phrases though)



PHRASE VS WORD BASED MT

- ▶ Phrase-pairs memorise:
 - ▶ common translation fragments (have access to **local context** in choosing lexical translation)
 - ▶ common reordering patterns (making up for naïve models of reordering)



SCORING PHRASES

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	black									
assumes		black	black	black	grey	grey				
that		grey	grey	grey	grey	black				
he							black			
will								pink	pink	dark pink
stay								pink	pink	dark pink
in								dark pink	pink	pink
the									pink	pink
house								pink	dark pink	pink

► Once we have all the phrase pairs

► compute counts over the whole corpus

► normalise counts to produce 'probabilities'

► E.g.,

Fig from Koehn09

$\phi(\text{im haus bleibt} | \text{will stay in the house})$

$$= \frac{c(\text{will stay in the house; im haus bleibt})}{c(\text{im haus bleibt})}$$

THE PHRASE-TABLE

- ▶ The **phrase-table** consists of all phrase-pairs and their scores, which forms the search space for decoding
- ▶ E.g., for ***natuerlich*** it may contain the following translation phrases

Translation	Probability $p(e f)$
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

- ▶ generally a massive list with many millions of phrase-pairs

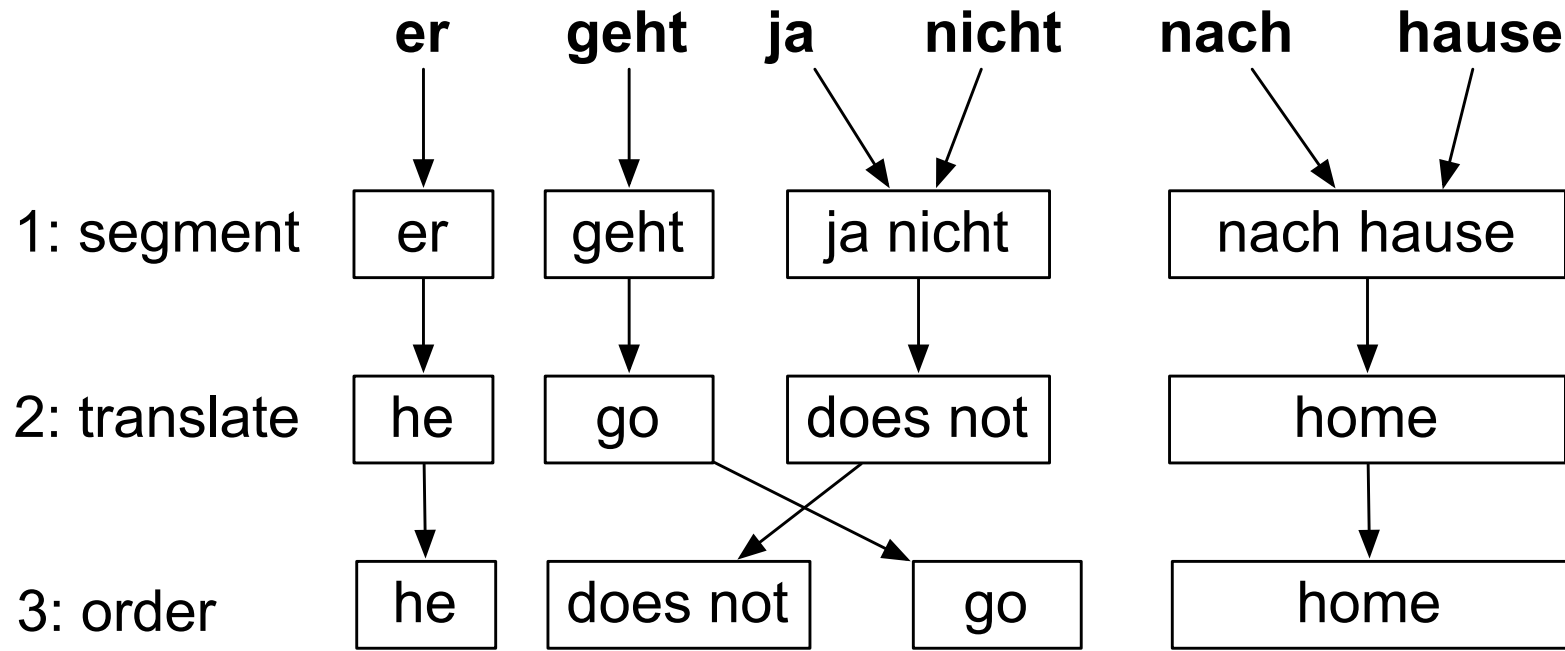
DECODING

$$E^*, A^* = \operatorname{argmax}_{E, A} \operatorname{score}(E, A, F)$$

- ▶ A describes the segmentation of F into phrases; and the re-ordering of their translations to produce E
- ▶ The *score* function is a product of the
 - ▶ translation “probability”, $P(F|E)$, split into phrase-pairs
 - ▶ language model probability, $P(E)$, over full sentence E
 - ▶ distortion cost, $d(\text{start}_i, \text{end}_{i-1})$, measuring amount of reordering between adjacent phrase-pairs
- ▶ Search problem
 - ▶ find translation E^* with the best overall score

TRANSLATION PROCESS

- Score the translations based on translation probabilities (step 2), reordering (step 3) and language model scores (steps 2 & 3).



SEARCH PROBLEM

er	geht	ja	nicht	nach	hause
he	is	yes	not	after	house
it	are	is	do not	to	home
, it	goes	, of course	does not	according to	chamber
, he	go	,	is not	in	at home
it is		not		home	
he will be		is not		under house	
it goes		does not		return home	
he goes		do not		do not	
	is		to		
	are		following		
	is after all		not after		
	does		not to		
	not				
	is not				
	are not				
	is not a				

- ▶ Cover all source words exactly once; visited in any order; and with any segmentation into “phrases”
- ▶ Choose a translation from phrase-table options

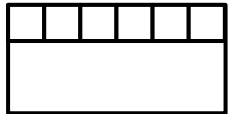
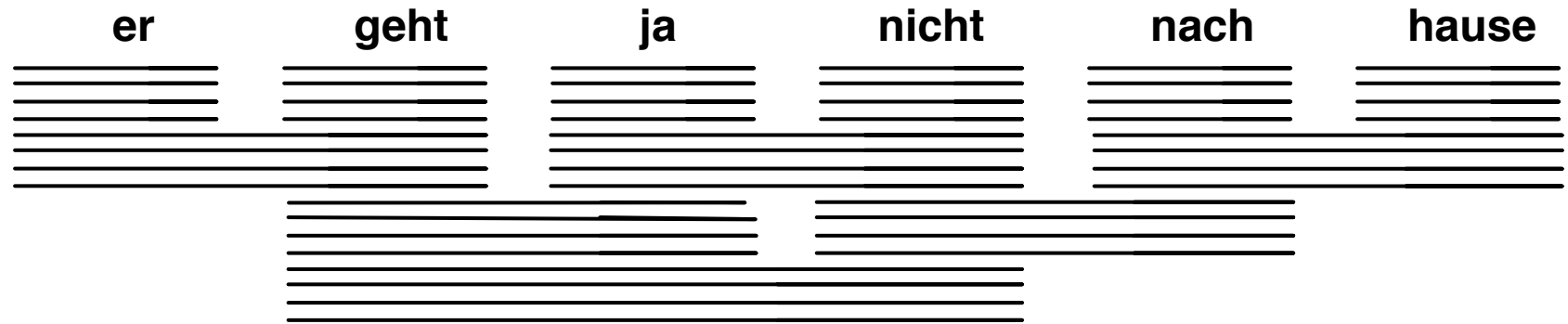
Leads to millions of possible translations...

Figure from Koehn, 2009

DYNAMIC PROGRAMMING SOLUTION

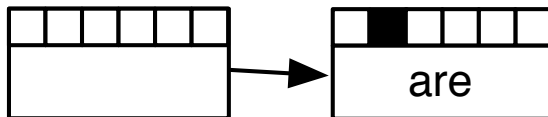
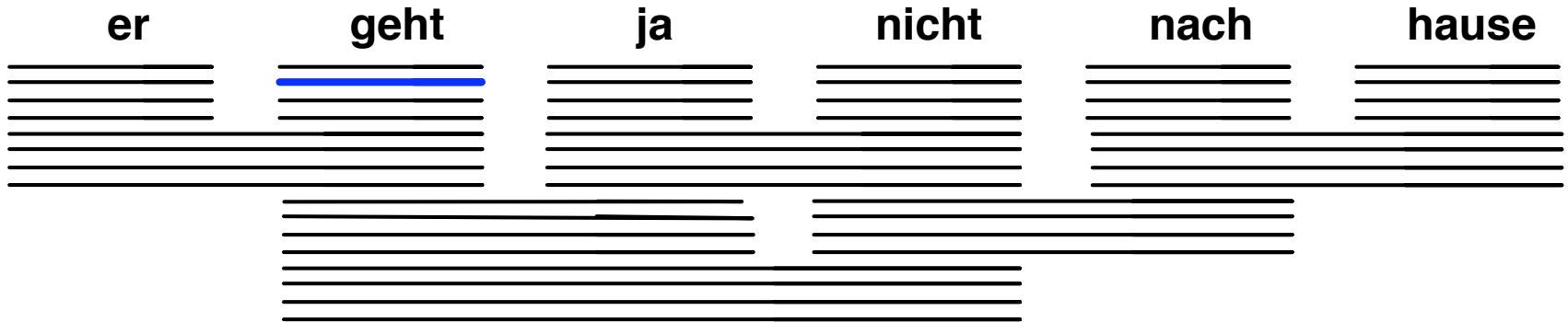
- ▶ Akin to Viterbi algorithm
 - ▶ factor out repeated computation
(like Viterbi for HMMs, “chart” used in parsing)
 - ▶ efficiently solve the maximisation problem
- ▶ Aim is to translate every word of the input once
 - ▶ searching over *every* segmentation into phrases;
 - ▶ the translations of each phrase; and
 - ▶ all possible ordering of the phrases

PHRASE-BASED DECODING



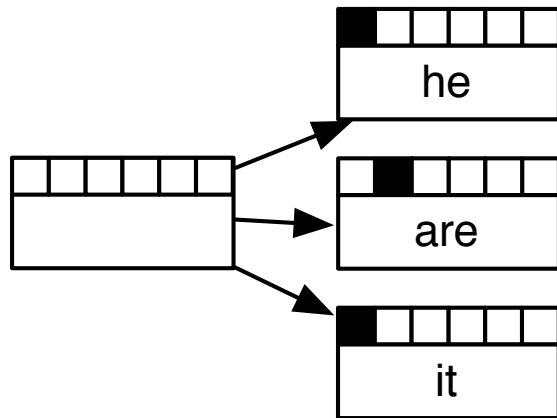
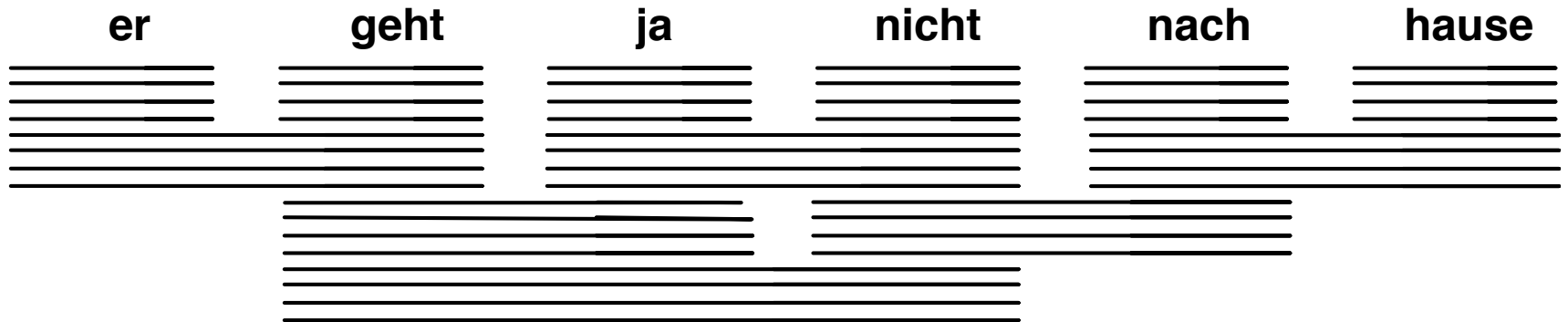
Start with empty state

PHRASE-BASED DECODING



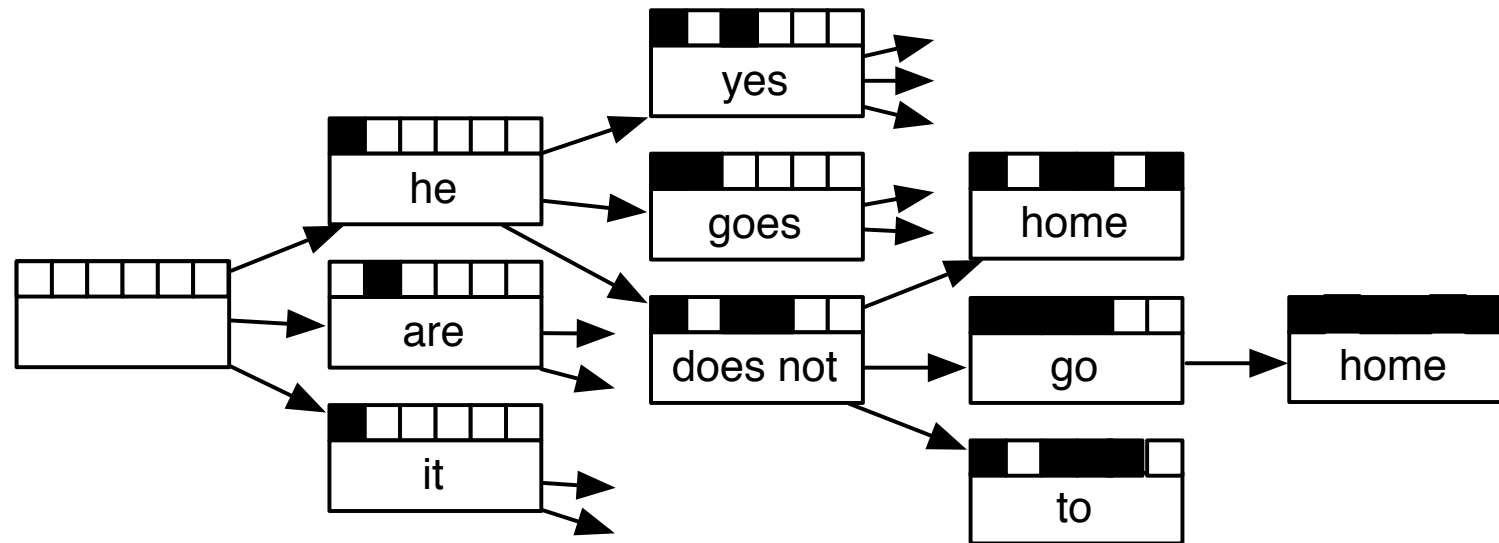
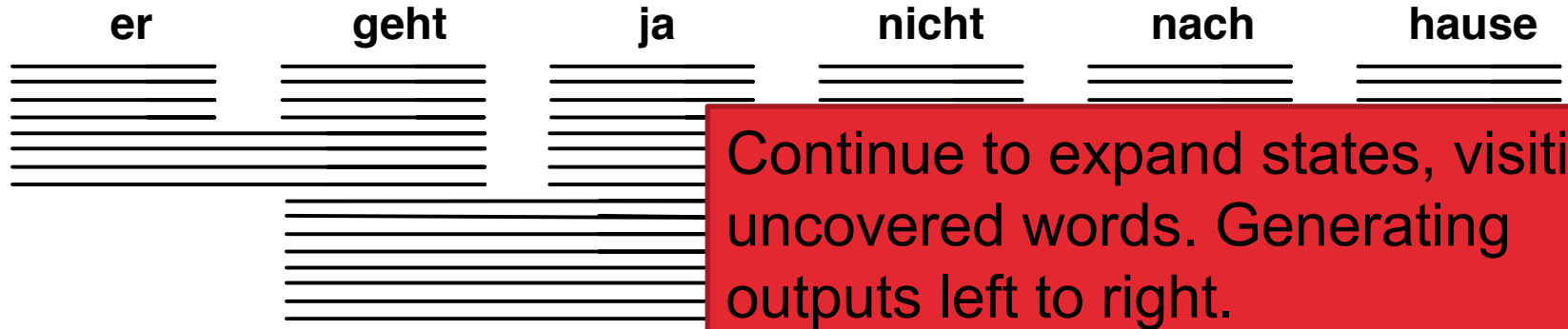
Expand by choosing
input span and
generating translation

PHRASE-BASED DECODING

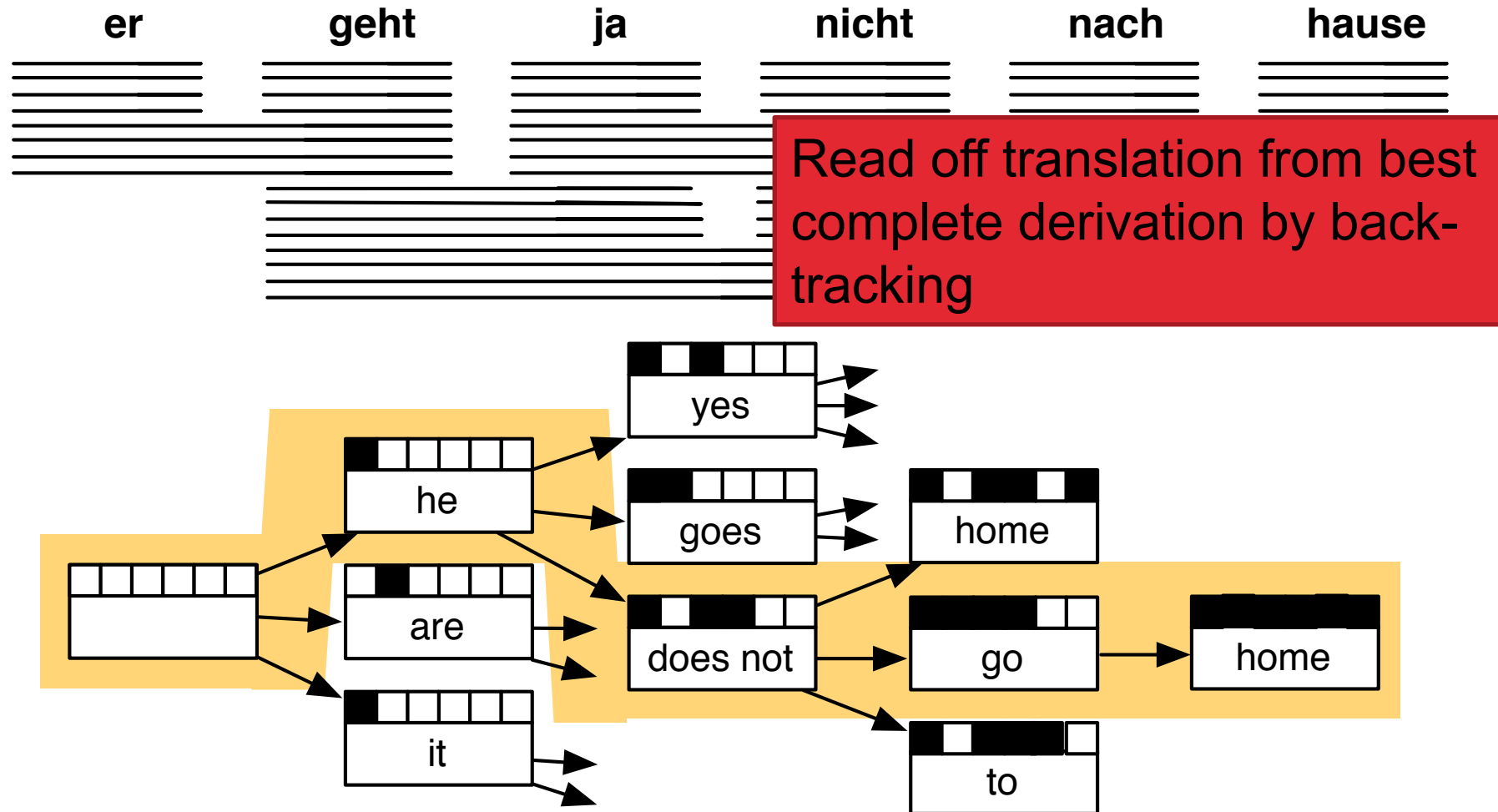


Consider all possible options to start the translation

PHRASE-BASED DECODING



PHRASE-BASED DECODING



REPRESENTING TRANSLATION STATE¹⁶

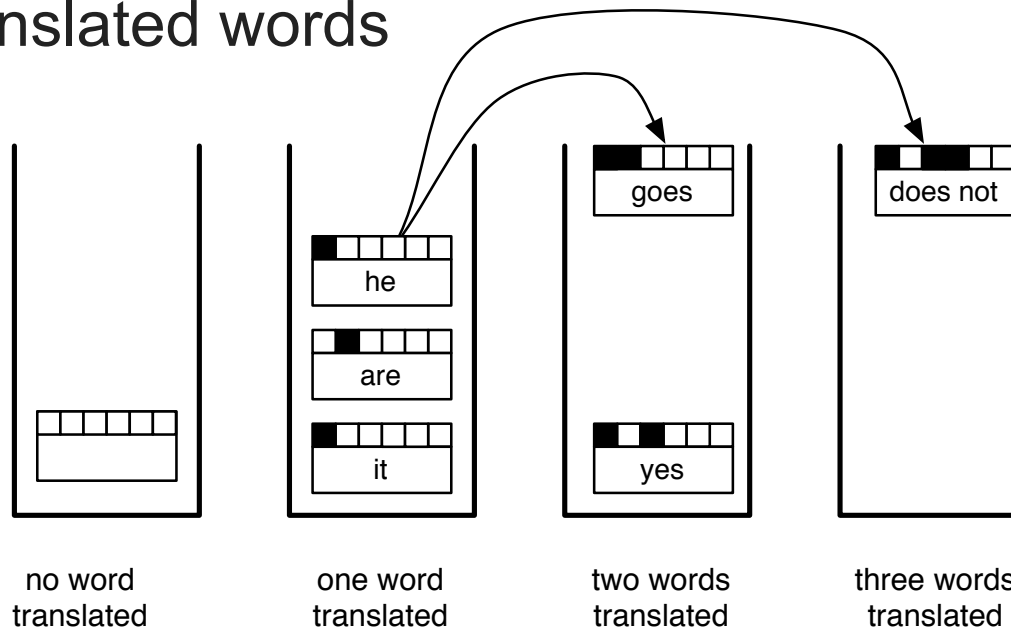
- ▶ Need to record
 - ▶ translation of phrase
 - ▶ which words are translated in bit-vector
 - ▶ last $n-1$ words in E... so that n gram LM can compute probability of subsequent words
 - ▶ end position of the last phrase translated in the source, for scoring distortion in next step
- ▶ Together allows for the score computation to be factorised

COMPLEXITY

- ▶ Full search is intractable
 - ▶ word-based and phrase-based decoding is NP complete (Knight 99)
 - ▶ arises from arbitrary reordering
- ▶ A solution is to prune the search space
 - ▶ Use **beam search**, a form of approximate search
 - ▶ maintaining no more than k options (“hypotheses”)
 - ▶ pruning over translations that cover a given number of input words

LENGTH BINNING & FUTURE COST

- ▶ Each time we extend a hypothesis, store resulting translation in bin according to source coverage
 - ▶ prune each bin to no more than k entries
 - ▶ also include approximate cost of translating the untranslated words



ADVANCED EXTENSIONS

▶ **More Features**

- ▶ often use many more than 3 features, although these are the central ones
- ▶ learn to weight the effect of each feature differently (MERT)

▶ **Grammars and trees**

- ▶ instead of just using phrase-pairs, can use pairs of CFG rules; parse F using one side of the translation grammar and then generate E using the other side

PHRASE-BASED MT SUMMARY

- ▶ Start with sentence-aligned parallel text
 - ▶ learn word alignments
 - ▶ extract phrase-pairs from word alignments
 - ▶ learn a language model
- ▶ Combine into decoding algorithm
 - ▶ ... and learn feature weights
- ▶ Apply to test sentences

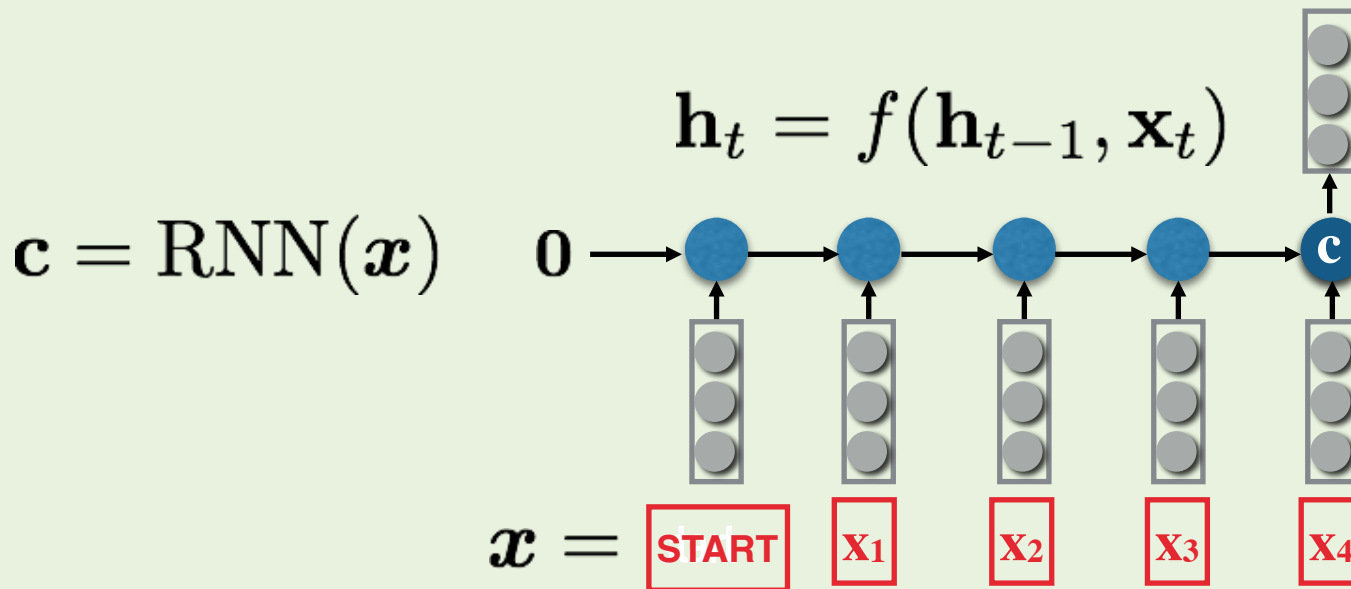
NEURAL MACHINE TRANSLATION

- ▶ Phrase-based approach rather complicated!
- ▶ Neural approach poses question:
 - ▶ Can we throw all this complexity away, instead learn a *single model* to directly translate from source to target?
- ▶ Using deep learning of neural networks
 - ▶ learn robust representations of words and sentences
 - ▶ attempts to generate words in the target given “deep” representation of the source

ENCODER-DECODER MODELS

- ▶ So-called “sequence2sequence” models combine:
 - ▶ **encoder** which represents the source sentence as a vector/matrix
 - ▶ akin to word2vec’s method for learning word vectors
 - ▶ **decoder** which predicts each word in the target
 - ▶ similar to a language model, except that the decoder is conditioned on the encoder representation
- ▶ Along with lots of CPU & GPU muscle...

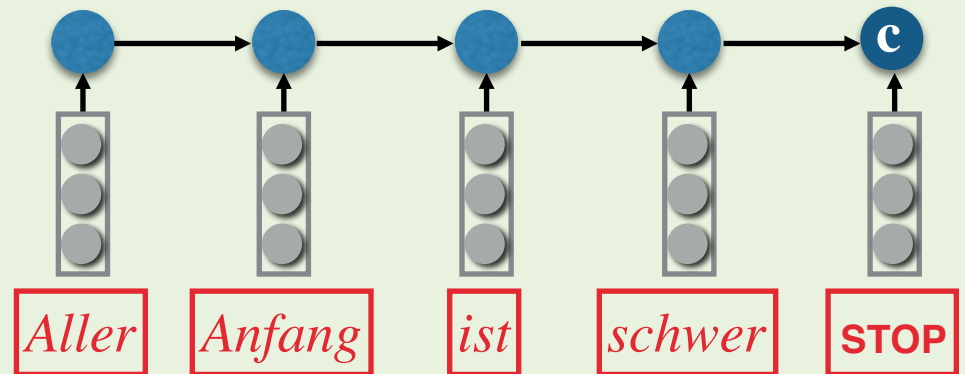
RECURRENT NEURAL NETWORKS (RNNS)



What is a vector representation of a sequence \mathbf{x} ?

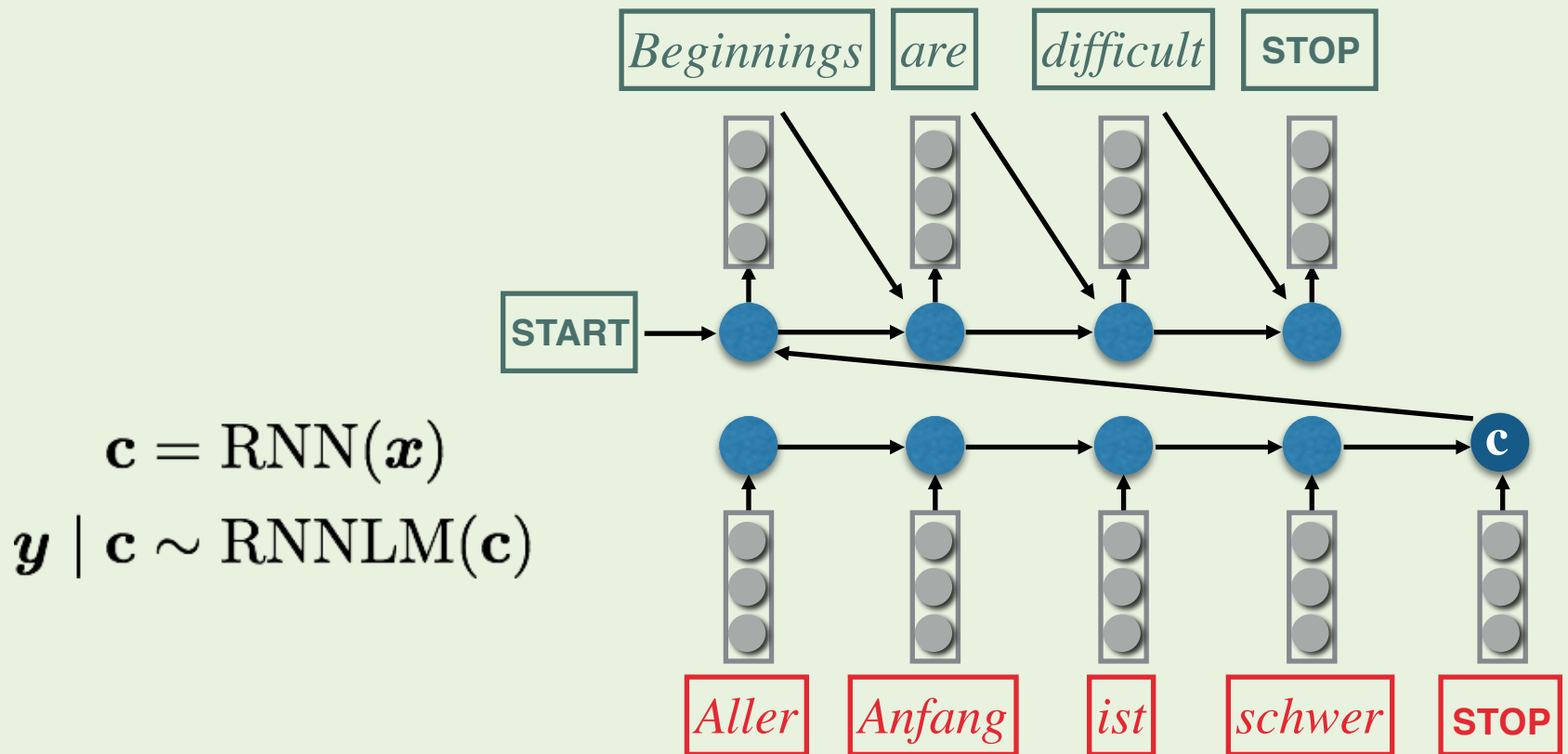
RNN ENCODER-DECODERS

$$\mathbf{c} = \text{RNN}(\mathbf{x})$$



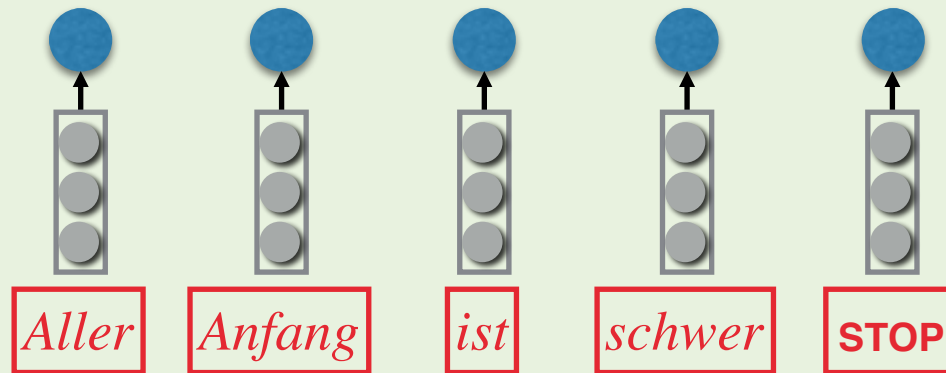
What is the probability of a sequence $\mathbf{y} \mid \mathbf{x}$?

RNN ENCODER-DECODERS



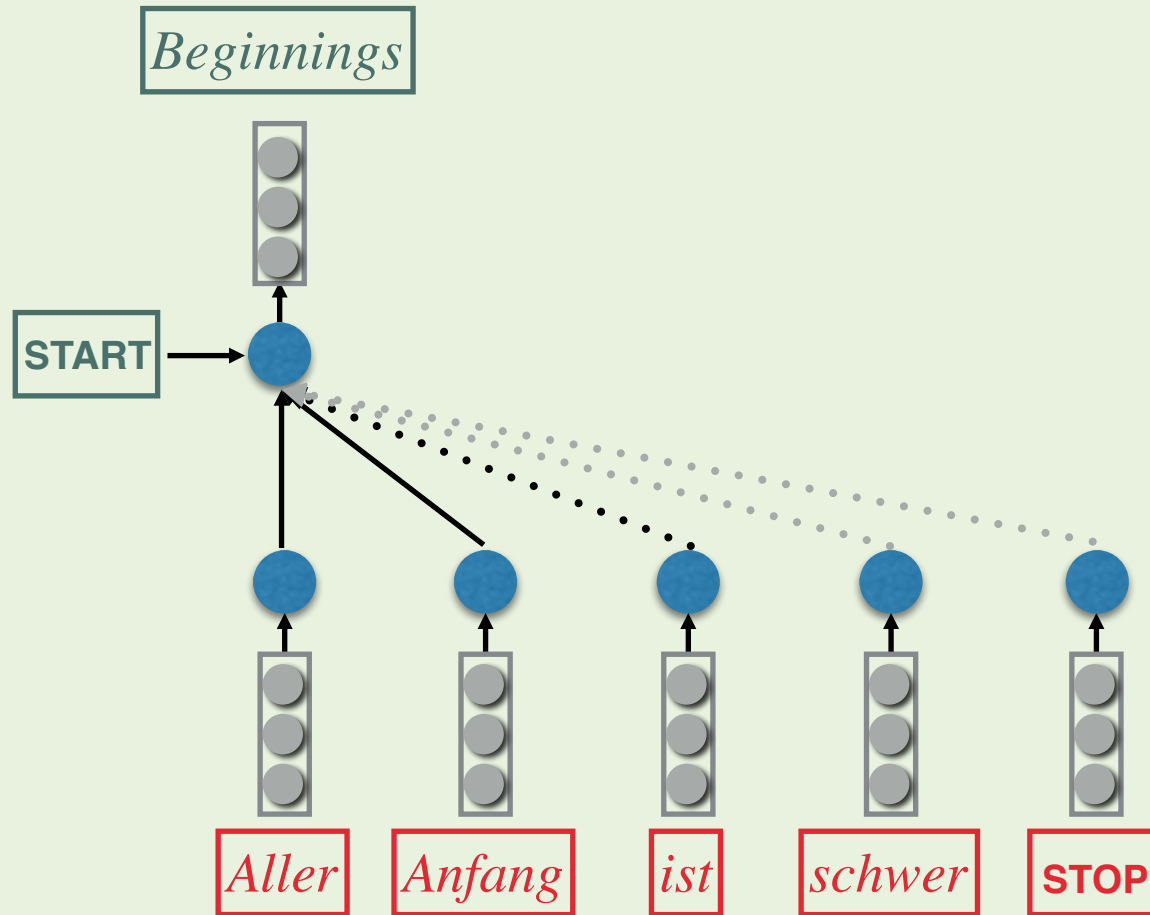
What is the probability of a sequence $\mathbf{y} \mid \mathbf{x}$?

RNN ATTENTION MODEL



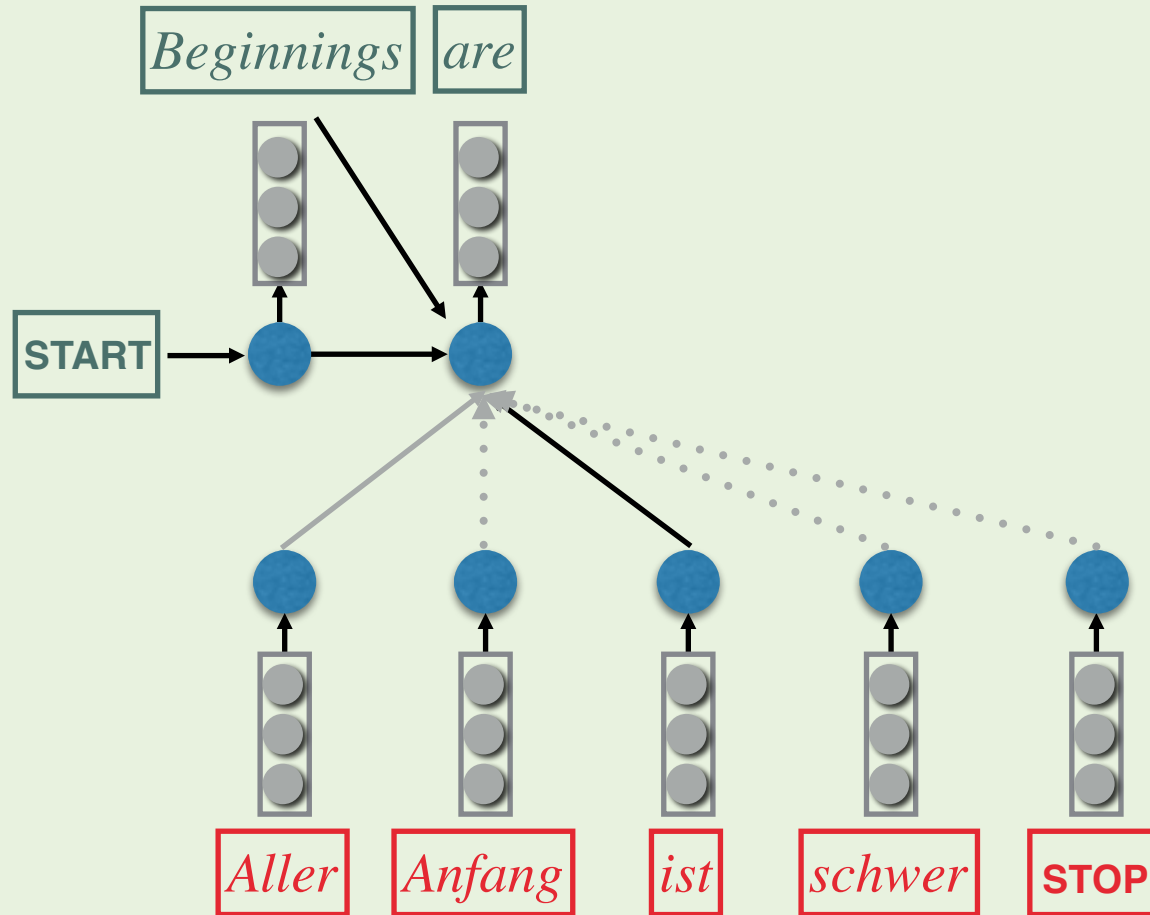
What is the probability of a sequence y | x ?

RNN ATTENTION MODEL



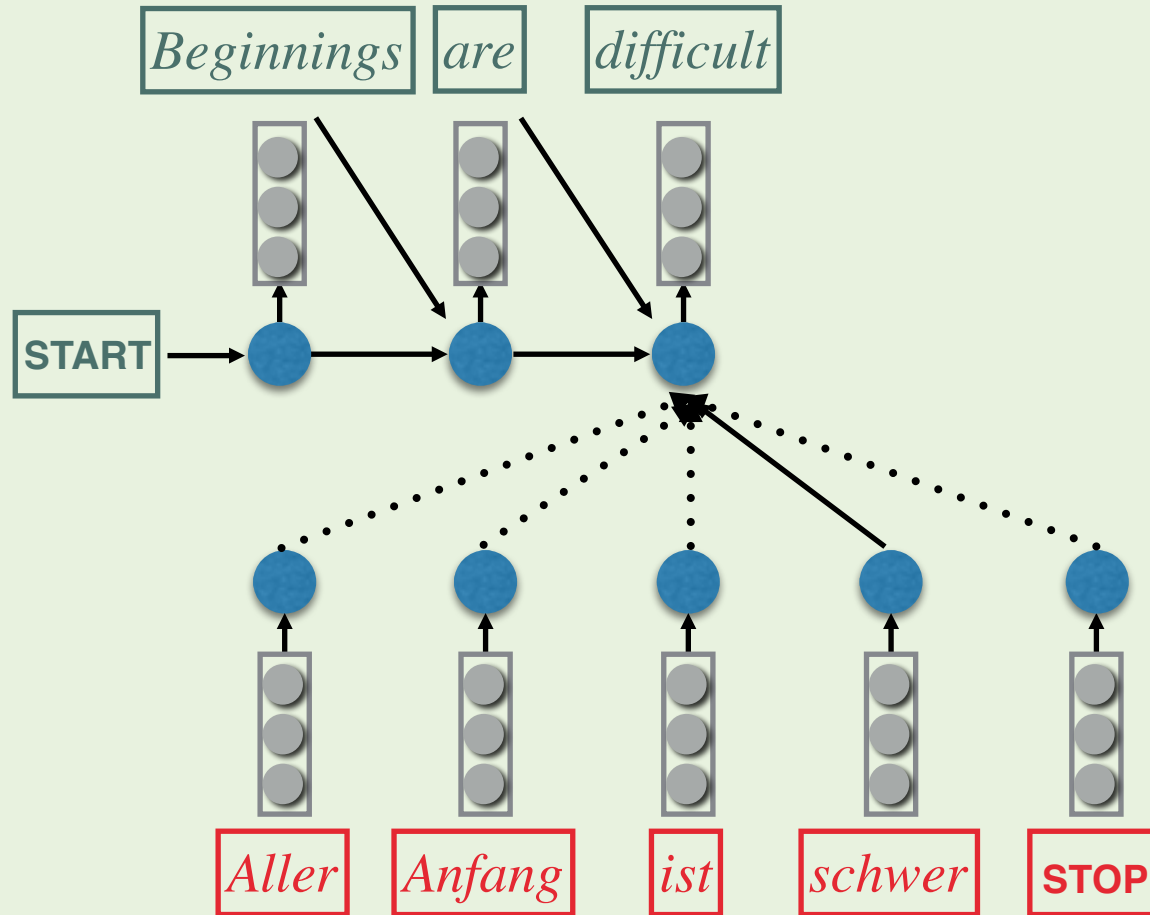
What is the probability of a sequence $y \mid x$?

RNN ATTENTION MODEL



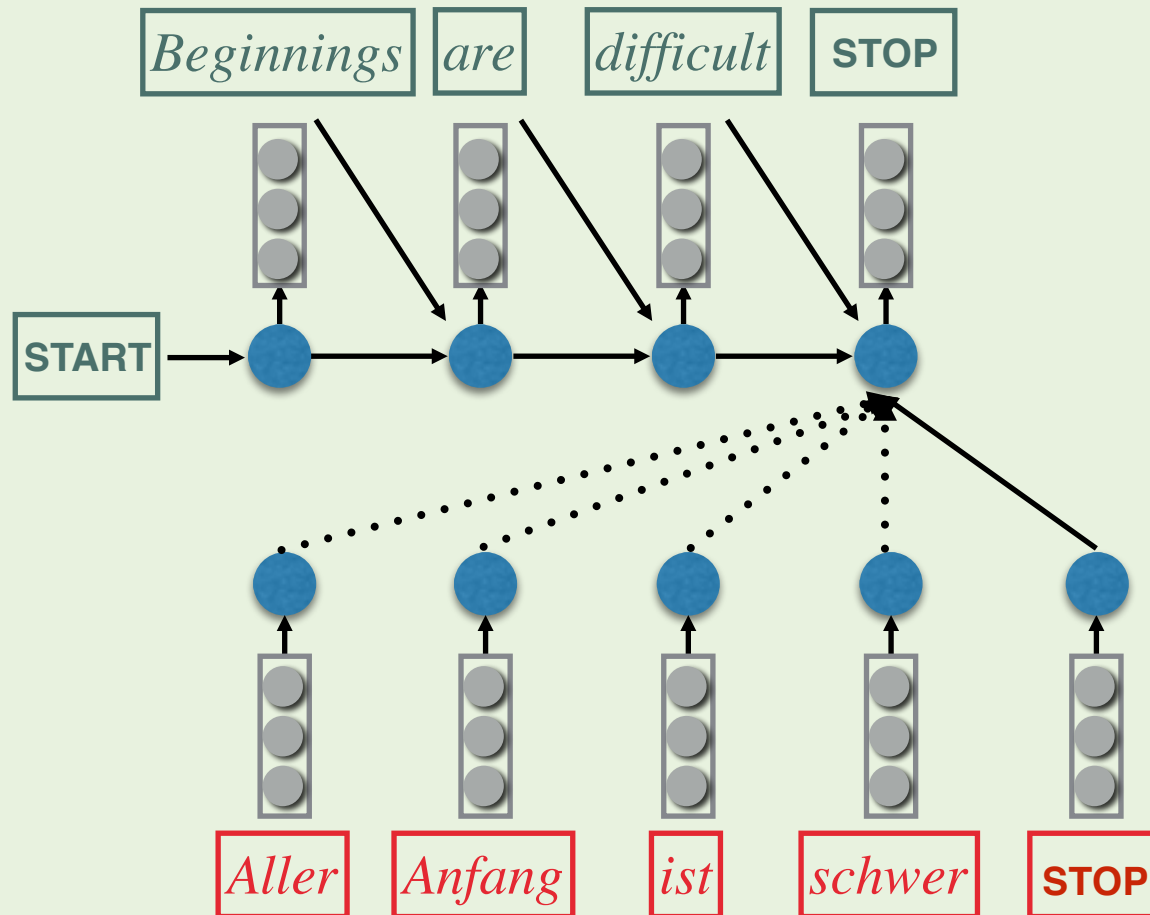
What is the probability of a sequence y | x ?

RNN ATTENTION MODEL



What is the probability of a sequence $y \mid x$?

RNN ATTENTION MODEL



What is the probability of a sequence $y \mid x$?

APPLICATIONS OF SEQ2SEQ

- ▶ Machine translation
- ▶ Summarisation (document as input)
- ▶ Speech recognition & speech synthesis
- ▶ Image captioning & image generation
- ▶ Word morphology (over characters)
 - ▶ e.g., study → student; receive → recipient;
play → player; pay → payer/payee
- ▶ Generating source code from text & more....

EVALUATION: DID IT WORK?

- ▶ Given input in Persian

ملبورن مهد و مرکز پیدی‌دای ش صنعت فی لمسازی و سی نما ، تلویزی ون ، رقص باله ، هنر امپرسی ون
 سبک‌های مختلف رقص مثل نی و وگ و ملبورن شافل در استرالی و مرکز مهم موز یک ک لاس یک و امروزی
 ن ک شو

- ▶ Google translate outputs the English

Melbourne cradle and center of origin of the film industry and cinema, television, ballet, art, impressionism, various dance styles such as New Vogue and the Melbourne Shuffle in Australia. Important center of classical and contemporary music in this country.

- ▶ We might ask a bilingual to judge, or compare to human translation

Melbourne is referred to as Australia's "cultural capital". It is the birthplace of Australian impressionism, Australian rules football, Australian film and television industries, and Australian contemporary dance such as the Melbourne Shuffle. It is recognised as a UNESCO City of Literature and a major centre for street art, music and theatre.

AUTOMATIC EVALUATION

- ▶ How many words are the shared between output:

Melbourne cradle **and** center **of** origin **of the film** industry **and** cinema, **television**, **street art**, **impressionism**, various **dance** styles **such as** New Vogue **and the Melbourne Shuffle** in **Australia and** an important center **of** classical and contemporary **music** in this country.

- ▶ And the reference:

Melbourne referred to **as Australia's** “cultural capital” it is the birthplace of Australian **impressionism**, Australian rules football, **the Australian film and television industry**, Australian contemporary **dance such as the Melbourne Shuffle**. It is recognized as a UNESCO City of Literature and a major **centre** for street **art, music** and the

MT EVALUATION: BLEU

- ▶ BLEU measures closeness of translation to one or more references

- ▶ defined as:

$$\text{BLEU} = \text{bp} \times \text{prec}_{1\text{-gram}} \times \text{prec}_{2\text{-gram}} \times \text{prec}_{3\text{-gram}} \times \text{prec}_{4\text{-gram}}$$

- ▶ weighted average of 1, 2, 3 & 4-gram precisions
 - ▶ $\text{prec}_{n\text{-gram}} = \text{num } n\text{-grams correct} / \text{num } n\text{-grams predicted in output}$
 - ▶ numerator clipped to #occurences of $n\text{gram}$ in the reference
 - ▶ and a brevity penalty to hedge against short outputs
 - ▶ $\text{bp} = \min (1, \text{output length} / \text{reference length})$
- ▶ Shown to have fair correlation with human judgements (also many other metrics: TER, METEOR, WER, ...)

SUMMARY

- ▶ Word vs phrase based MT
 - ▶ Components of phrase-base approach
 - ▶ Decoding algorithm
- ▶ Neural encoder-decoder
- ▶ Reading
 - ▶ JM2 25.7 – 25.9
 - ▶ Koehn09 5.1 – 5.2 and 6.1 – 6.2
 - ▶ **JFF: Neural Machine Translation and Sequence-to-sequence Models: A Tutorial**, Neubig 2017
<https://arxiv.org/abs/1703.01619>