

AI för naturligt språk

Introduktion till textgenerering

Marco Kuhlmann

Institutionen för datavetenskap

<

>

talktotransformer.com

Private Research Teaching LiU

+

Talk to Transformer

See how a modern neural network completes your text. Type a custom snippet or try one of the examples. [Learn more](#) below.

 Follow @AdamDanielKing

 for more neat neural networks.

Custom prompt

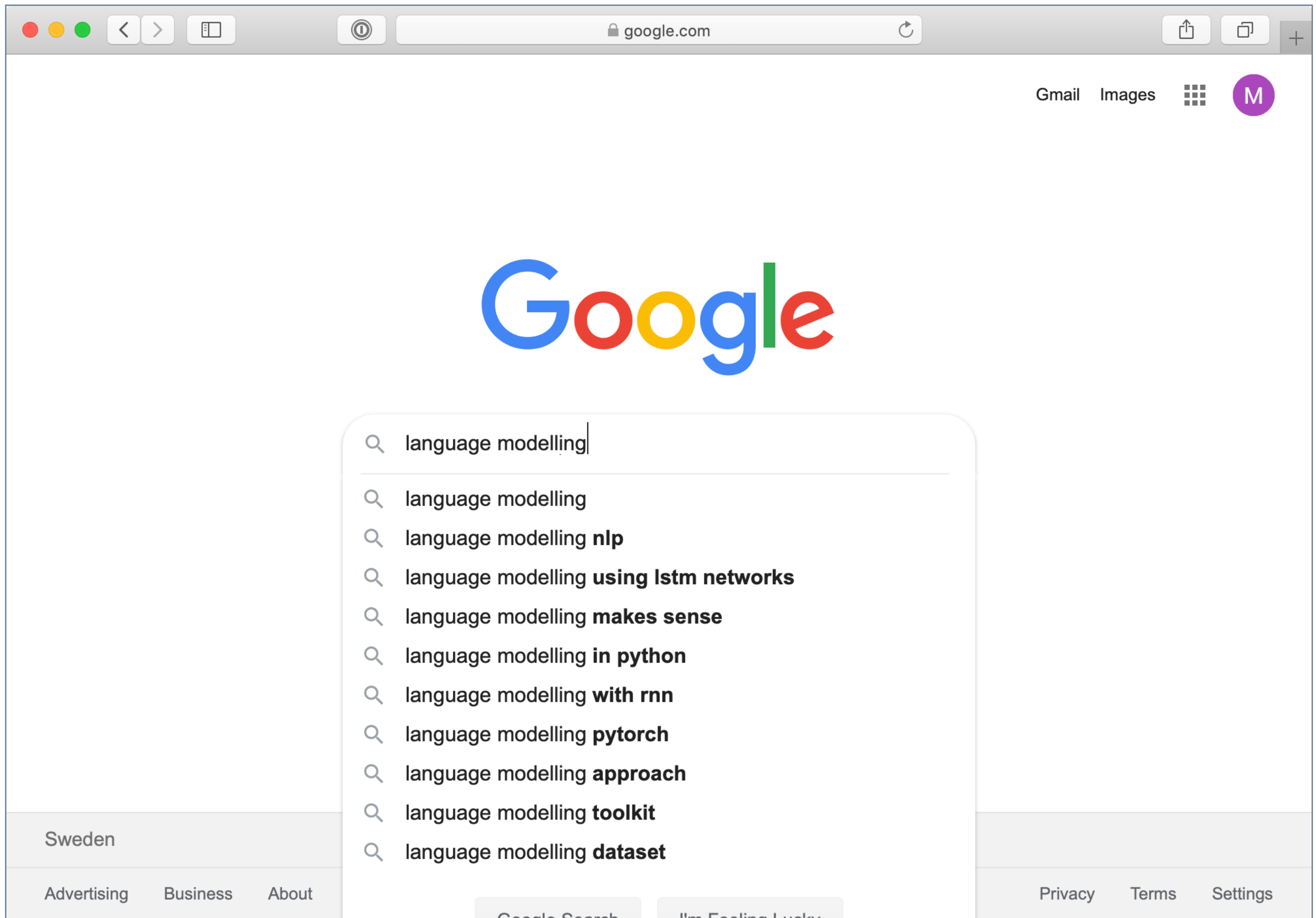
Kim hates broccoli.

GENERATE ANOTHER

Completion

Kim hates broccoli. They're making her sick. Well, it looks like the other week she had a flashback and went into the freezer where all of the vegetables used to be and she has never seen a broccoli plant before, so she never knew that if you kept it out of the cold it would magically grow to enormous size. Eventually she saw that it grew. So basically, thanks to a self-imposed dorm shortage in the lower four-fours, I have recently arranged to fill the highest-

 Privacy - Terms



Att generera text, ett ord i taget

- Ett sätt att formalisera textgenerering är att se det som en uppgift där vi slumpar fram meningar ord för ord.
- Vi tänker oss att vi redan har en sekvens av ord $w_1, \dots, w_t$ och nu vill veta sannolikheten för alla möjliga nästa ord, w_{t+1} :

$$P(w_{t+1} \mid w_1, \dots, w_t)$$

- Här antar vi som vanligt att w_{t+1} kommer från en fix vokabulär; vi kommer alltså aldrig att hitta på helt nya ord.

Textgenerering och språkmodeller

- Våra sannolikheter för ”nästa ord” kan även användas för att tilldela sannolikheter till redan färdiga meningar.

Hur sannolikt är det att den här texten är skriven på svenska, franska?

- Dessa två perspektiv är ekvivalenta, eftersom sannolikheten för en ordsekvens kan skrivas i termer av betingade sannolikheter: *

$$P(w_1 \cdots w_N) = \prod_{t=1}^N P(w_t \mid w_1, \dots, w_{t-1})$$

* Det finns vissa tekniska detaljer som vi bortser ifrån här.

Språkmodeller för översättning

Rangordna möjliga översättningar för följande mening:

他 向 记者 介绍了 主要 内容

han för reporter presenterade huvudsaklig innehåll

- Han för reportern presenterade det huvudsakliga innehållet.
- Han presenterade det huvudsakliga innehållet för reportern.
- Han presenterade huvudinnehållet för reportern.

AI för naturligt språk

N-gram-modeller

Marco Kuhlmann

Institutionen för datavetenskap

N-gram-modeller

- Ett ***n*-gram** är en sekvens av *n* stycken ord (eller tecken).
unigram (*den*), bigram (*den sista*), trigram (*den sista delen*)
- En ***n*-gram-modell** är en språkmodell baserad på *n*-gram – en sannolikhetsfördelning över sekvenser av *n* ord.
- I samband med textgenerering är vi intresserade sannolikheten för det sista ordet i ett *n*-gram, givet de föregående orden:

$$P(w_n \mid w_1 \cdots w_{n-1})$$

Unigram-modeller

En **unigram-modell** är samma sak som bag of words-modellen:

$$P(w_1 \cdots w_N) = \prod_{i=1}^N P(w_i)$$

I en sådan modell är alltså alla ordsannolikheter parvist oberoende av varandra.

Bigram-modeller

I en **bigram-model** ser faktoriseringen av sannolikheten ut så här:

$$P(w_1 \cdots w_N) = P(w_1 \mid \text{beginning-of-sentence}) \cdot \prod_{i=2}^N [P(w_i \mid w_{i-1})] \cdot P(\text{end-of-sentence} \mid w_N)$$

I en sådan modell är alltså sannolikheten för varje ord endast beroende av det närmast föregående ordet.

Bigram-sannolikheten för en ordsekvens

beginning-of-sentence

$$P(abc) = P(a \mid \text{BOS}) \cdot P(b \mid a) \cdot P(c \mid b) \cdot P(\text{EOS} \mid c)$$

end-of-sentence

Två sätt att använda n-gram-modeller

- **Predicering**

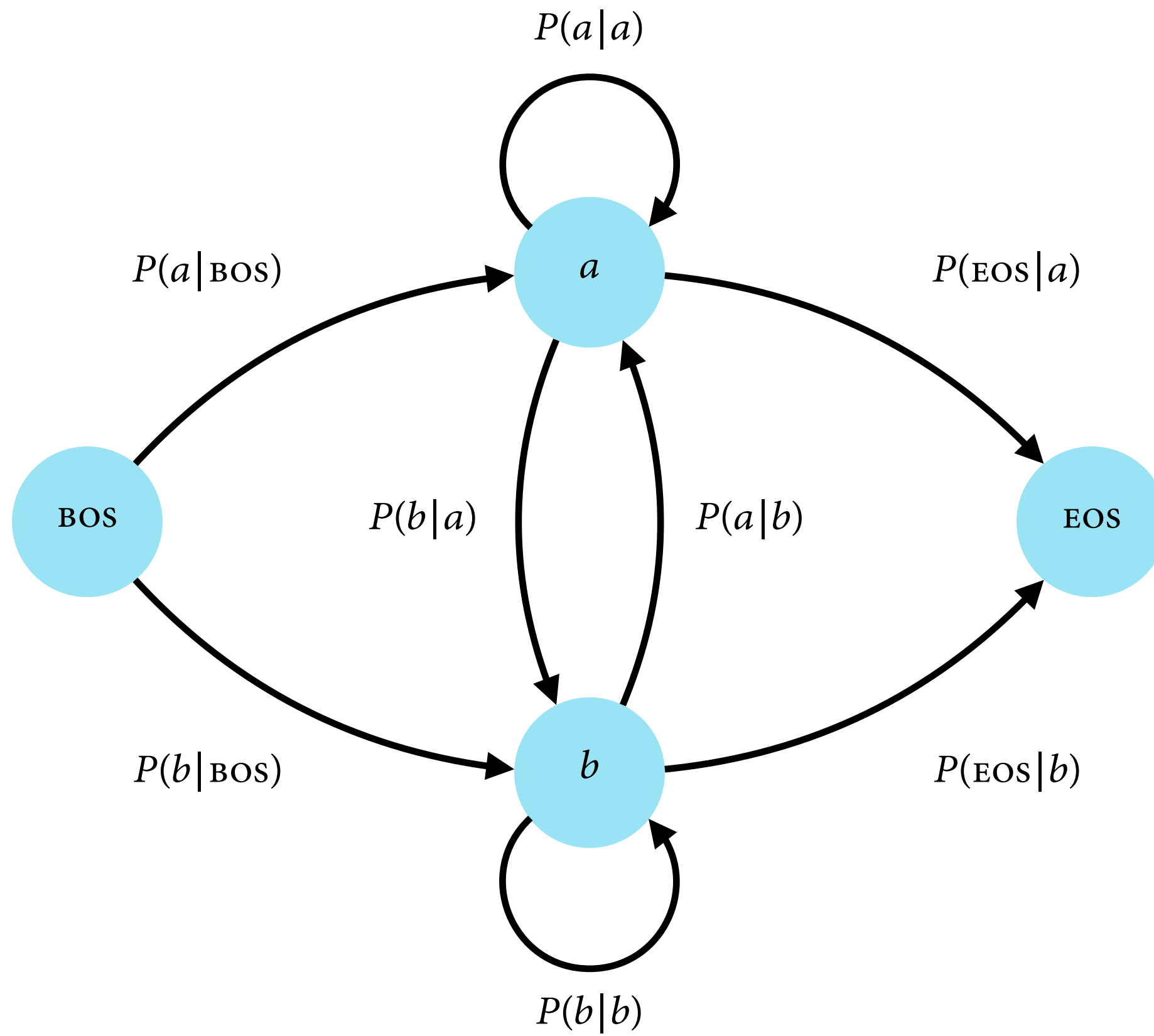
För att predicera nästa ord tar vi det ord som har högst sannolikhet bland alla möjliga nästa ord w :

predicerat ord = $\operatorname{argmax}_w P(w | \text{föregående ord})$

- **Generering**

För att generera en text slumpar vi fram det nästa ordet w enligt sannolikheten $P(w | \text{föregående ord})$.

Detta genererar inte nödvändigtvis meningsfulla texter.



AI för naturligt språk

Att skatta n-gram-modeller

Marco Kuhlmann

Institutionen för datavetenskap

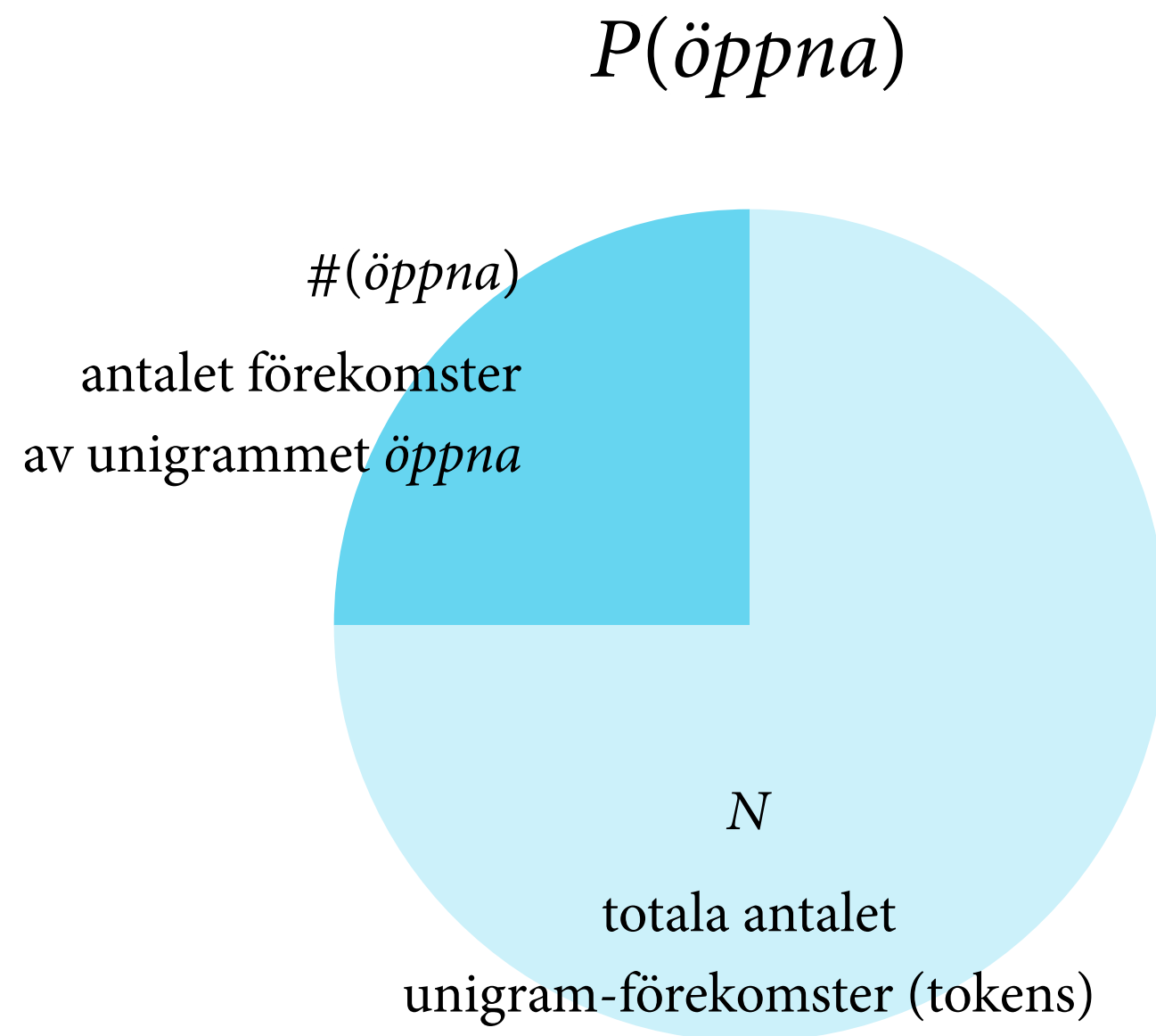
Att lära sig n -gram-modeller

- Det enklaste sättet att lära sig n -gram-modeller är att använda sig av maximum likelihood estimation (MLE).

samma princip som i Naive Bayes

- Den generella principen är att vi vill hitta parametrar (i detta fall: sannolikheter) som maximerar sannolikheten av datat.
- I samband med n -gram-modeller kan vi helt enkelt bara räkna antalet förekomster av n -gram och normalisera.

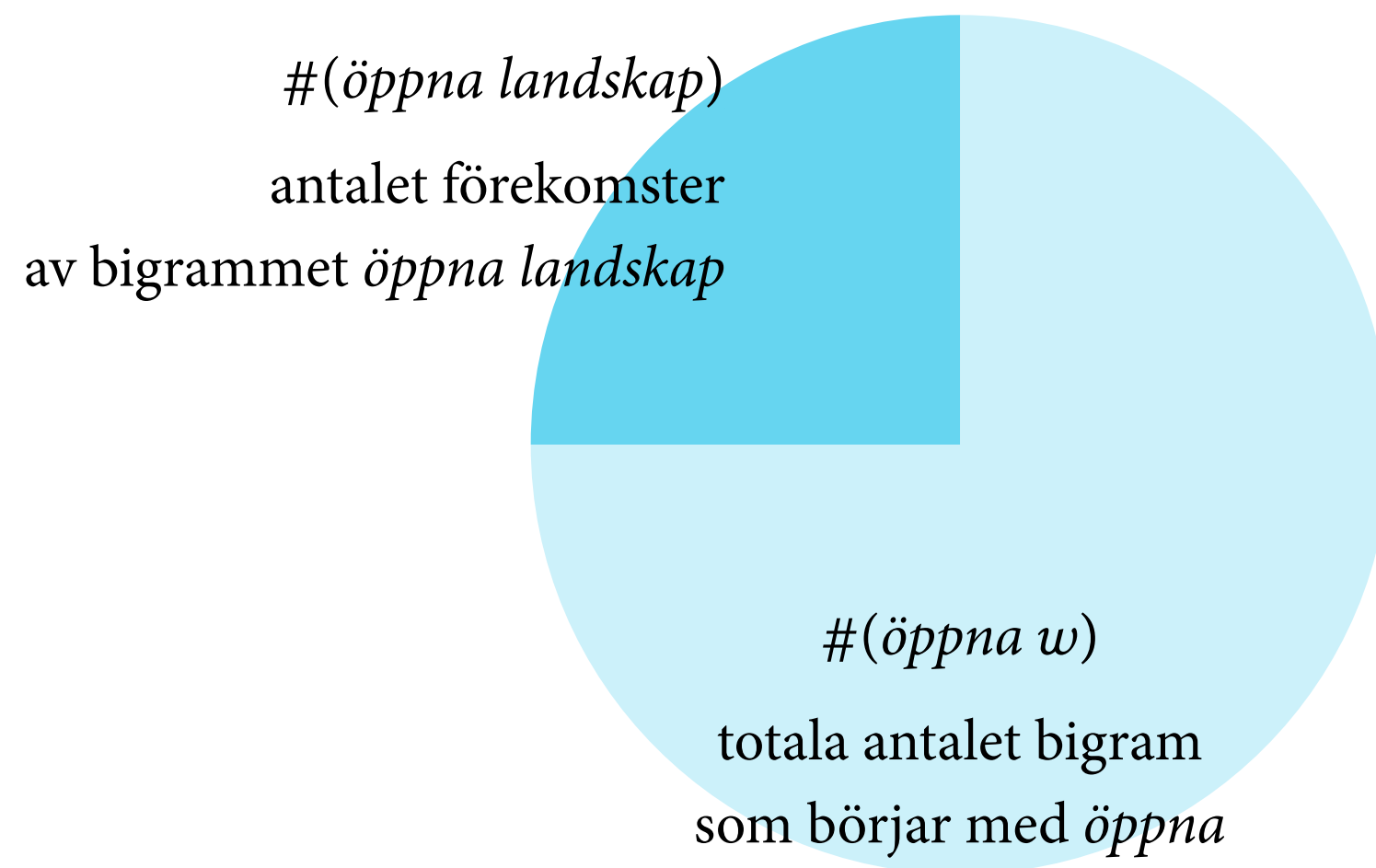
MLE av unigram-sannolikheter



$$P(w) = \frac{\#(w)}{N}$$

MLE av bigram-sannolikheter

$$P(\textit{landskap} | \textit{\textit{öppna}})$$



$$P(w | u) = \frac{\#(uw)}{\#(u\bullet)}$$

$$P(w | u) = \frac{\#(uw)}{\#(u)}$$

Problem med datagleshet

Vad händer om *students opened their w*
inte förekommer i datan?
Då har w sannolikhet 0.

Möjliga lösningar:
utjämning, discounting

$$P(w \mid \text{students opened their}) = \frac{\#(\text{students opened their } w)}{\#(\text{students opened their})}$$

Vad händer om *students opened their*
inte förekommer i datan?
Då är sannolikheten odefinierad!

Möjliga lösningar:
back-off, interpolering

AI för naturligt språk

Neurala språkmodeller

Marco Kuhlmann

Institutionen för datavetenskap

Begränsningar med n -gram-modeller

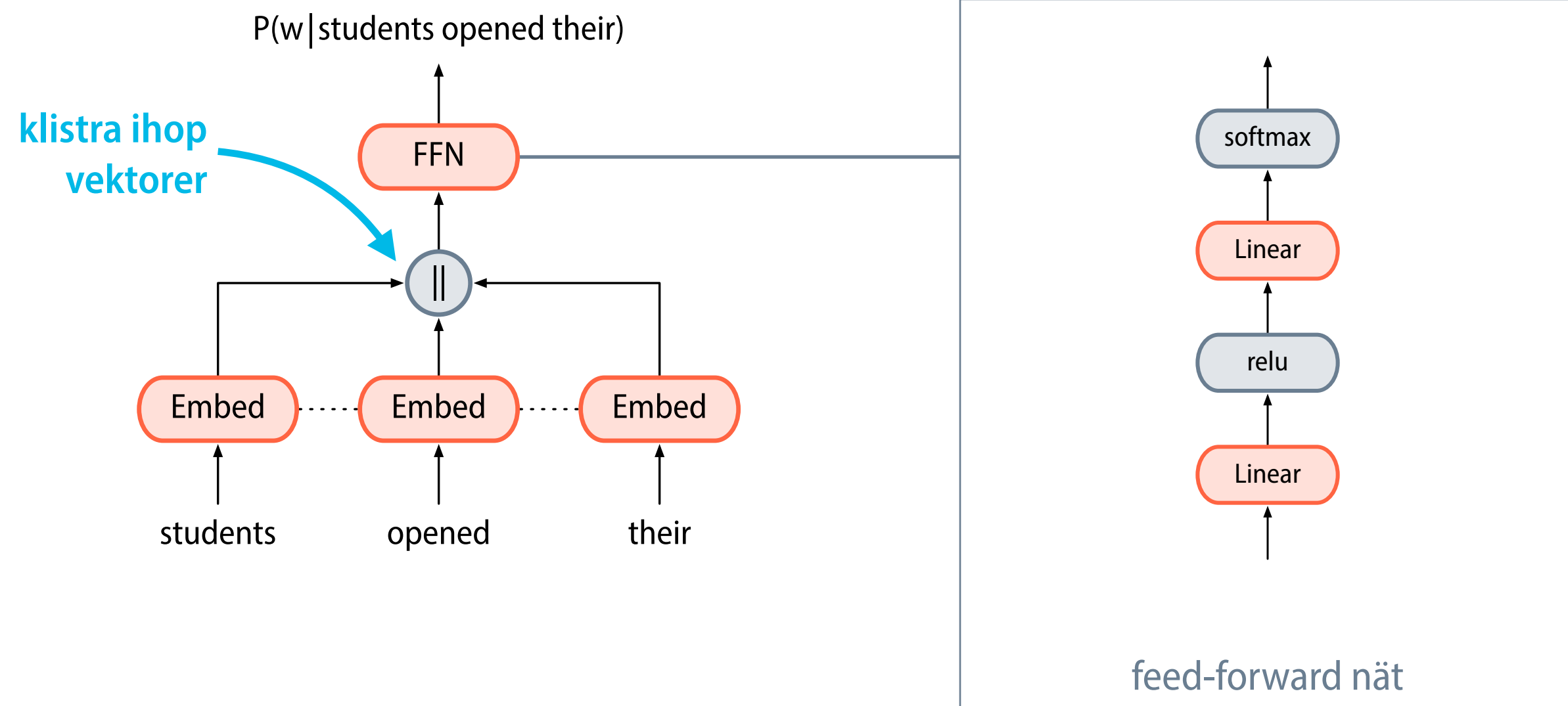
Goldberg § 9.3.2

- Att skala upp till större n -gram-storlekar är problematiskt på grund av både beräkningsproblem och växande datagleshet.
- Skattning av bra n -gram-modeller kräver relativt komplicerade tekniker såsom utjämning och backoff.

Att observera rött äpple påverkar inte automatiskt skattningar för grönt äpple.

- Att generera text ”ur tomma intet” är inte särskilt användbart; vi behöver ha en representation av vad vi vill generera.

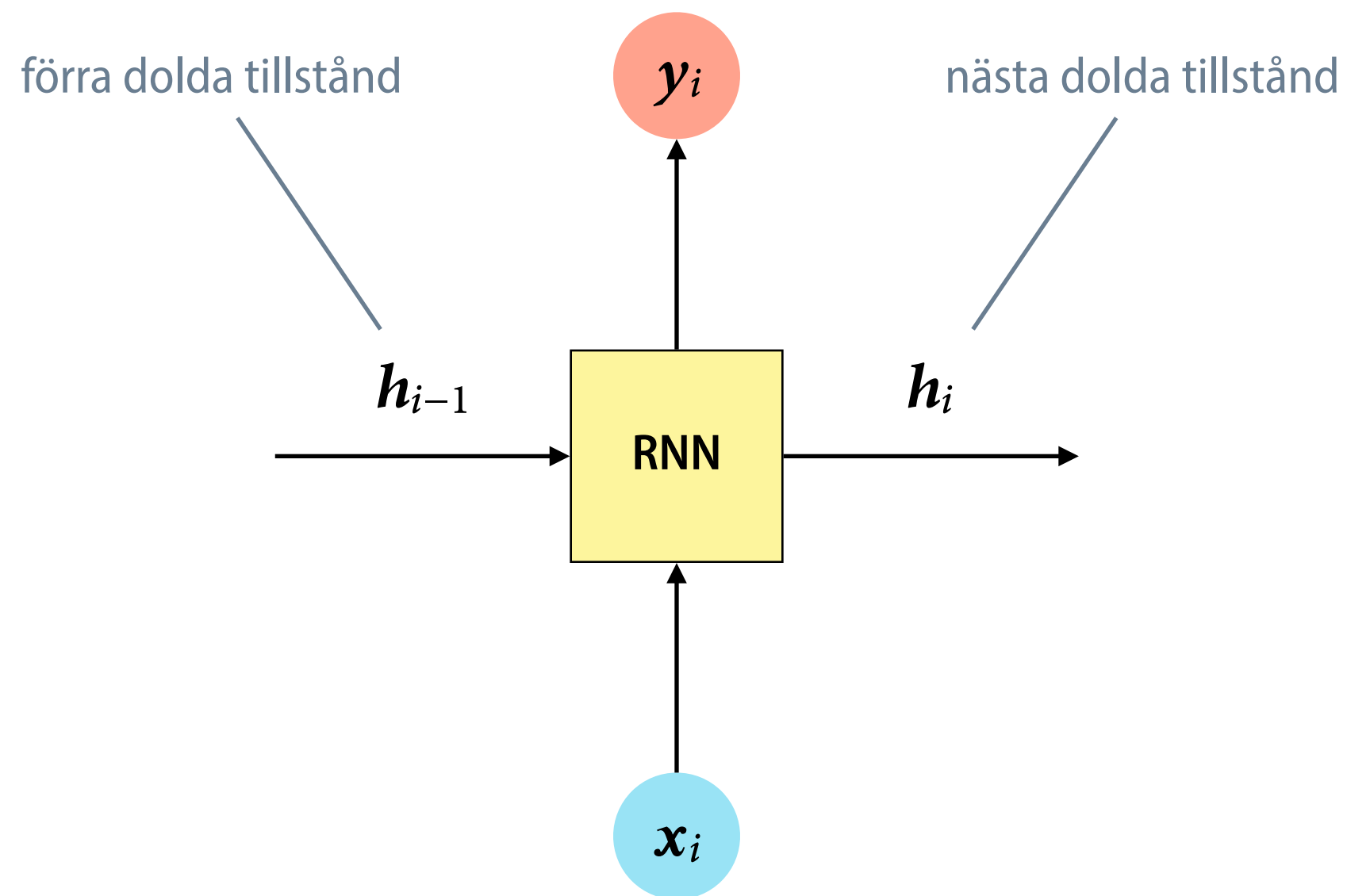
Textgenerering med fönstermodellen



Rekurrenta neurala nät

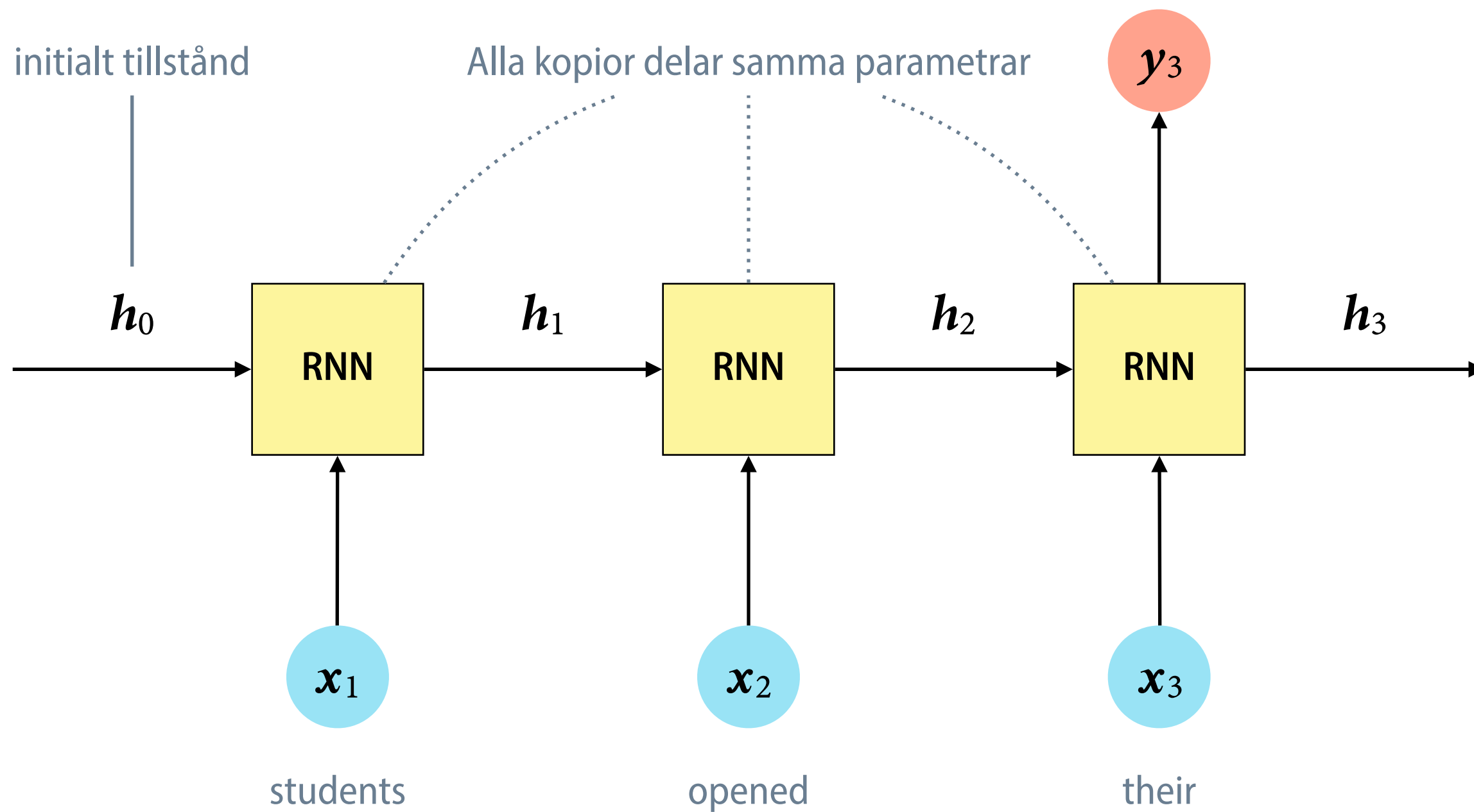
- I motsats till fönstermodellen kan **rekurrenta neurala nät (RNNs)** processa obegränsat långa sekvenser av ord.
- När ett RNN ska processa en given sekvens så ”vecklar man ut” det genom att kopiera nätet till varje position i sekvensen.
- Alla kopior delar samma uppsättning parametrar.

RNN, rekursivt perspektiv



$$h_i = H(h_{i-1}, x_i) \quad y_i = O(h_{i-1}, x_i)$$

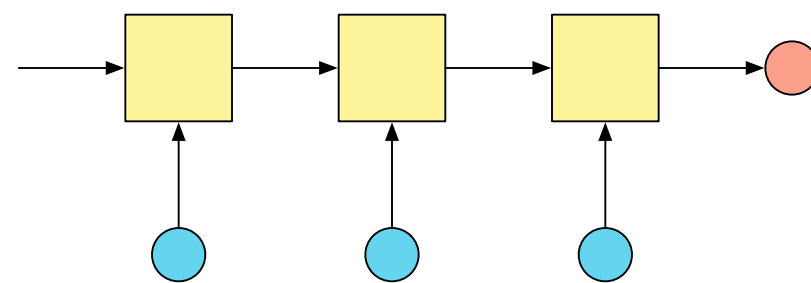
RNN, utvecklat



Egenskaper hos rekurrenta neuronnät

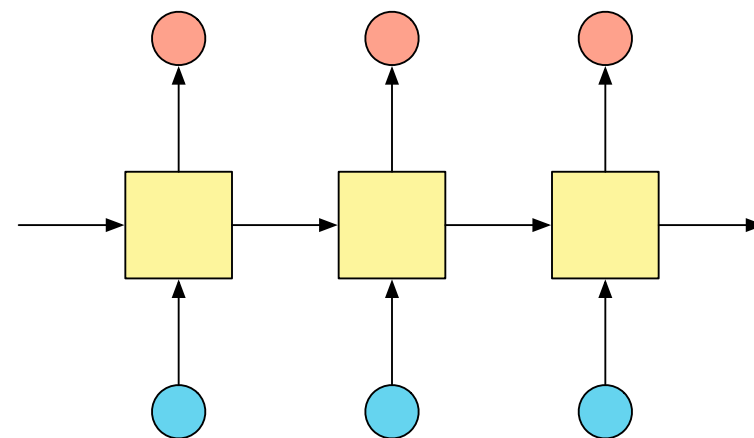
- Alla positionsspecifika kopior delar alla modellparametrar.
Antalet parametrar växer inte med sekvenslängden.
- Outputen kan påverkas av hela sekvensen hittills.
Jämför detta med antagandet i fönstermodellen.
- Sluttillståndet kan ses som en ”sammanfattning” av inputen.
Det kan representera en användbar information om input-sekvensen.

Typiska tillämpningar av RNNs



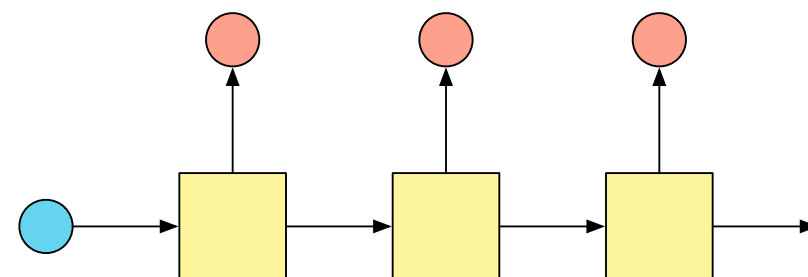
encoder

exempel: textklassificering



transducer

exempel: ordklasstagging



decoder

exempel: textgenerering

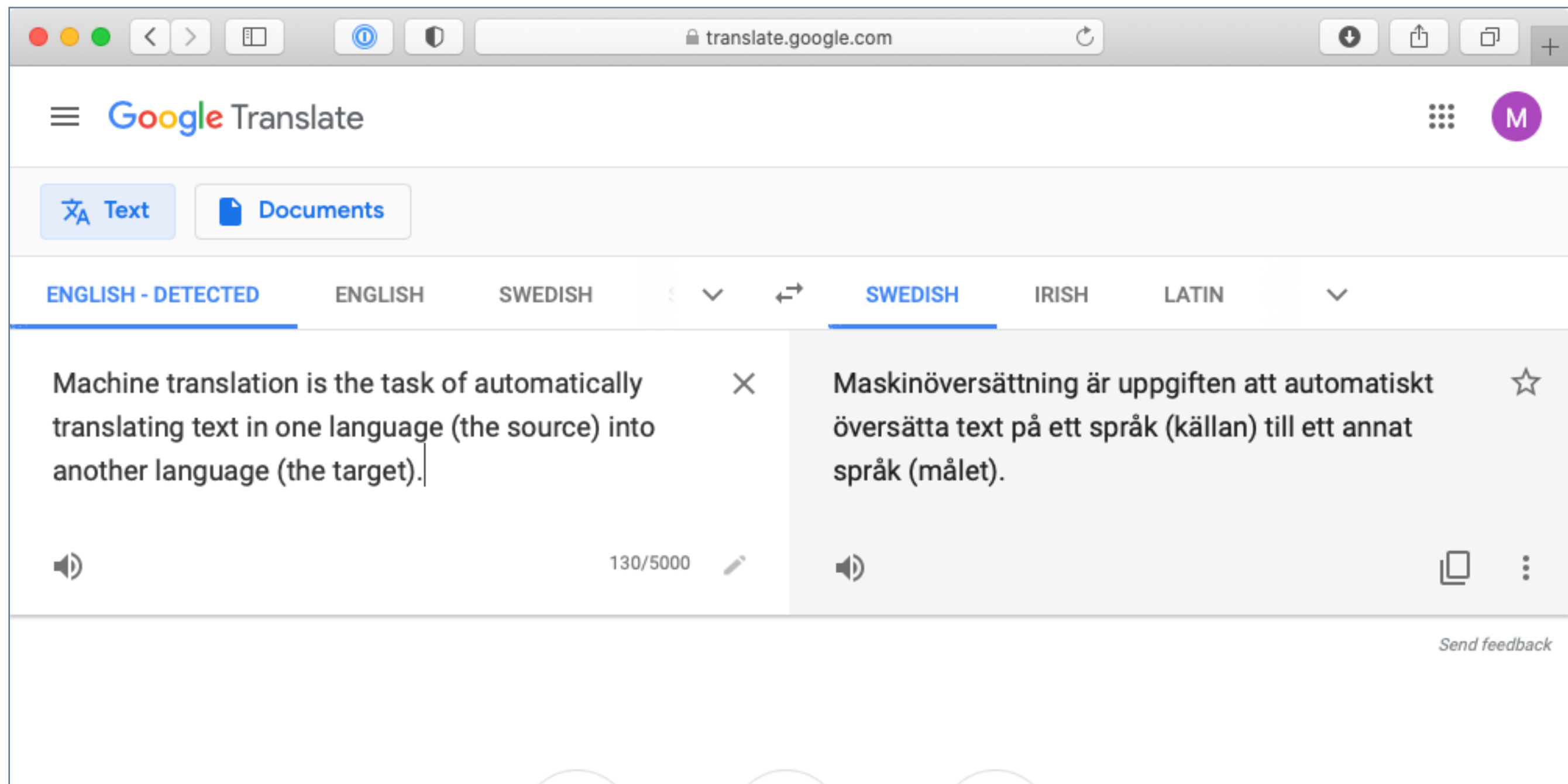
AI för naturligt språk

Maskinöversättning

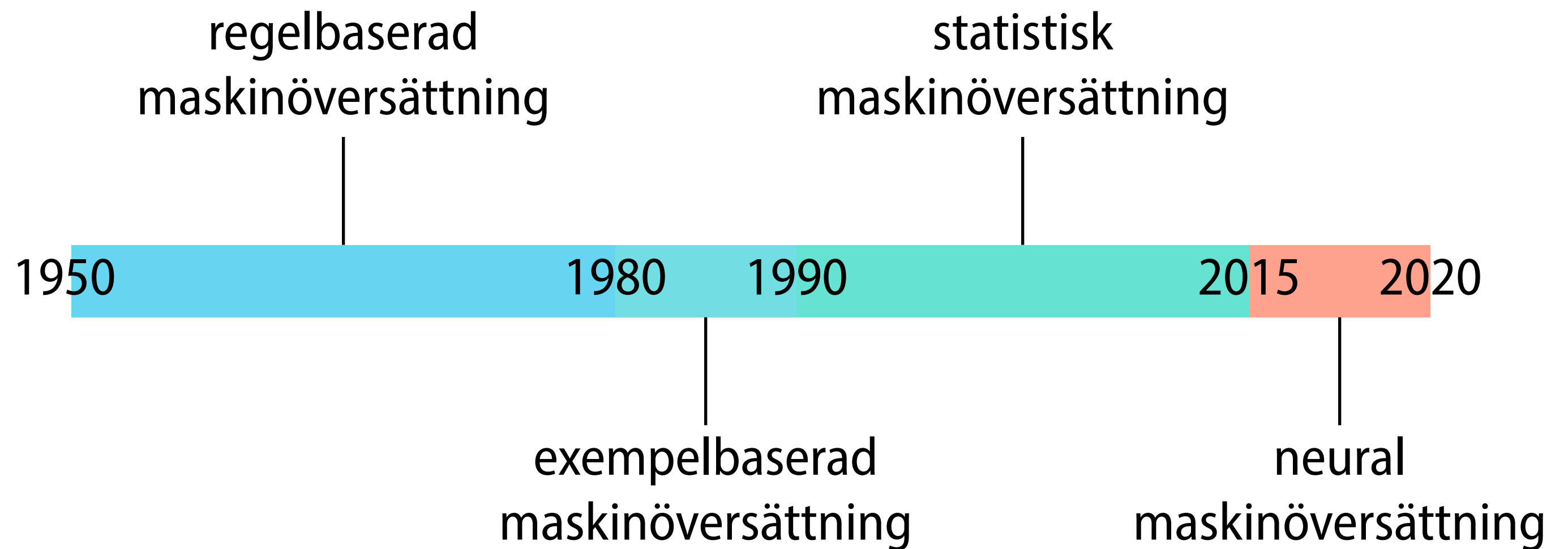
Marco Kuhlmann

Institutionen för datavetenskap

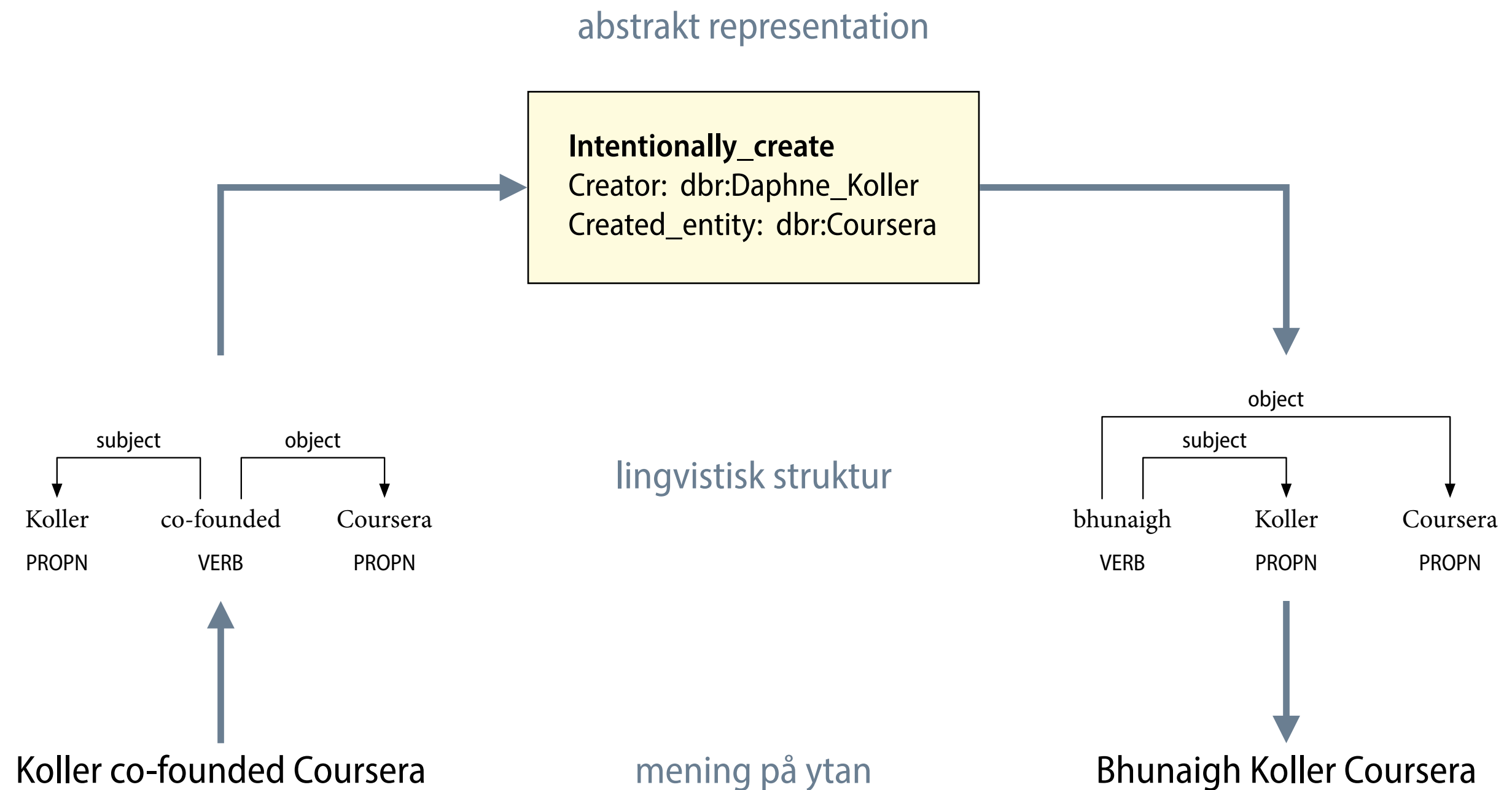
Maskinöversättning



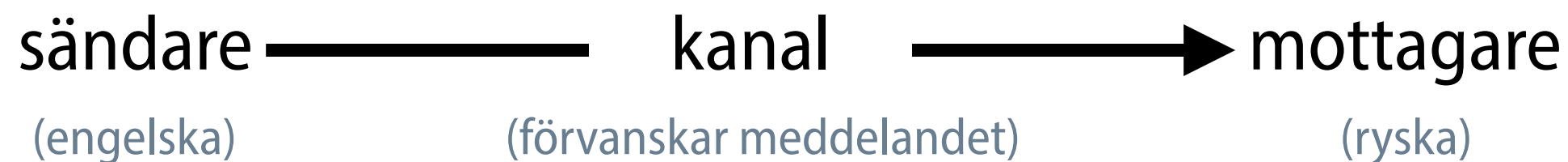
En tidslinje



Ett klassiskt perspektiv på maskinöversättning



Den brusiga kanalen



Bildkälla



When I look at an article in Russian, I say:
‘This is really written in English,
but it has been coded in some strange symbols.
I will now proceed to decode.’

Warren Weaver (1894–1978)

AI för naturligt språk

Neural maskinöversättning

Marco Kuhlmann

Institutionen för datavetenskap

Neural maskinöversättning

- **Neural maskinöversättning** modellerar översättningsuppgiften genom ett enda neurala nät.
- De första systemen för neural maskinöversättning använde rekurrenta nät; idag använder de flesta Transformers.

Sekvens till sekvens-modellen (seq2seq)

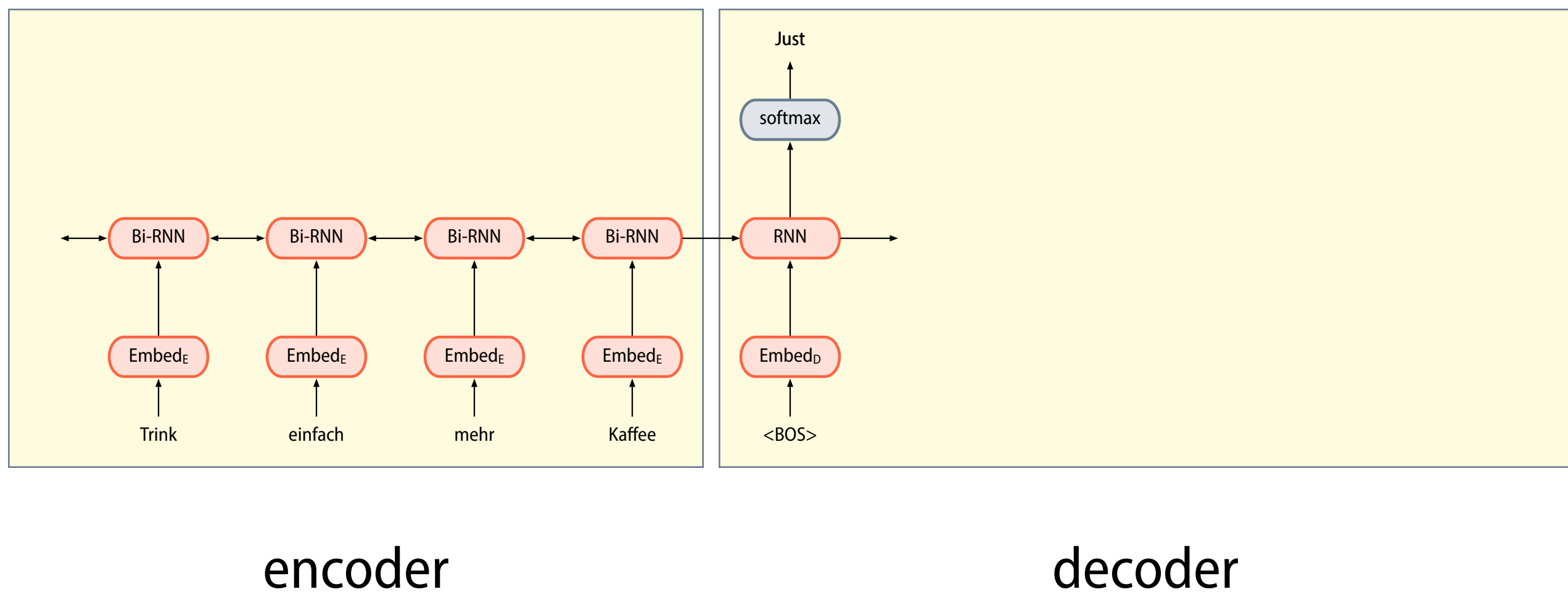
Sekvens till sekvens-modellen består av två komponenter:

- En nätkomponent som producerar en representation av meningen på källspråket – **encoder**.
- En nätkomponent som tar in encoders output och genererar meningen på målspråket – **decoder**.

typiskt ett (bidirektionellt) rekurrent neuronnät

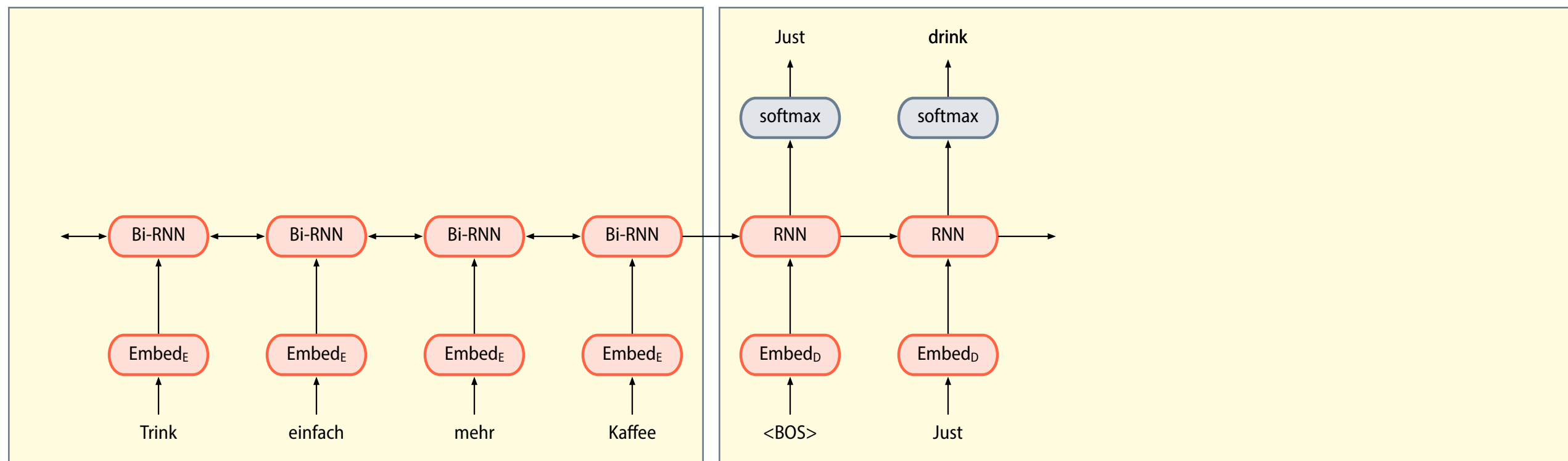
autoregressiv = genererar ett ord i taget

Standard seq2seq-architekturen



[Sutskever et al. \(2014\)](#)

Standard seq2seq-arkitekturen

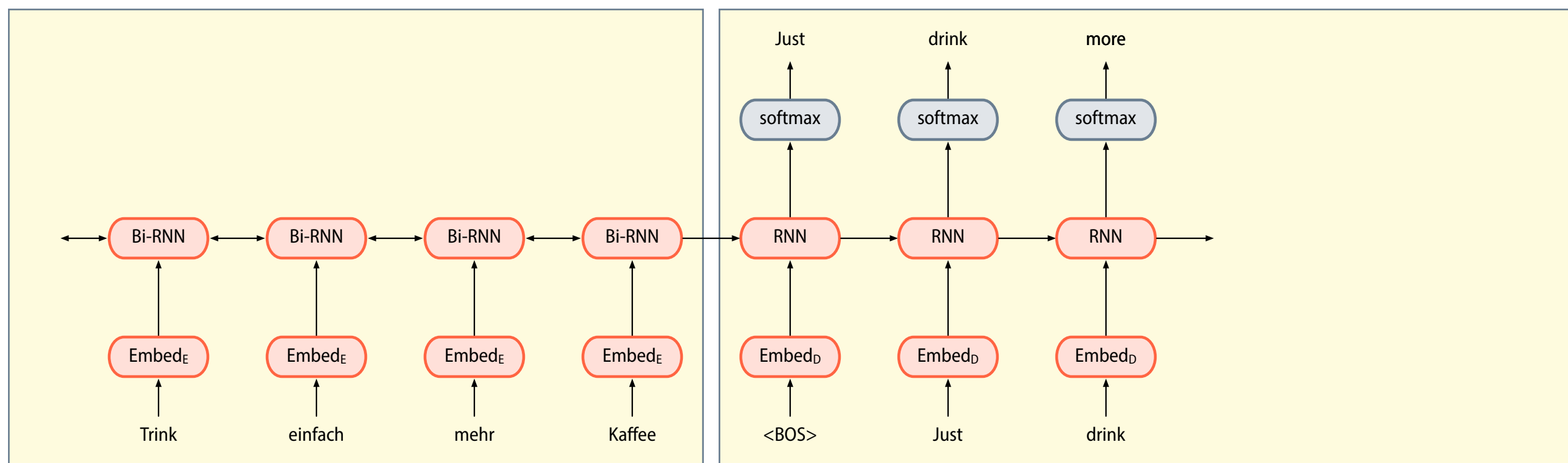


encoder

decoder

[Sutskever et al. \(2014\)](#)

Standard seq2seq-architekturen

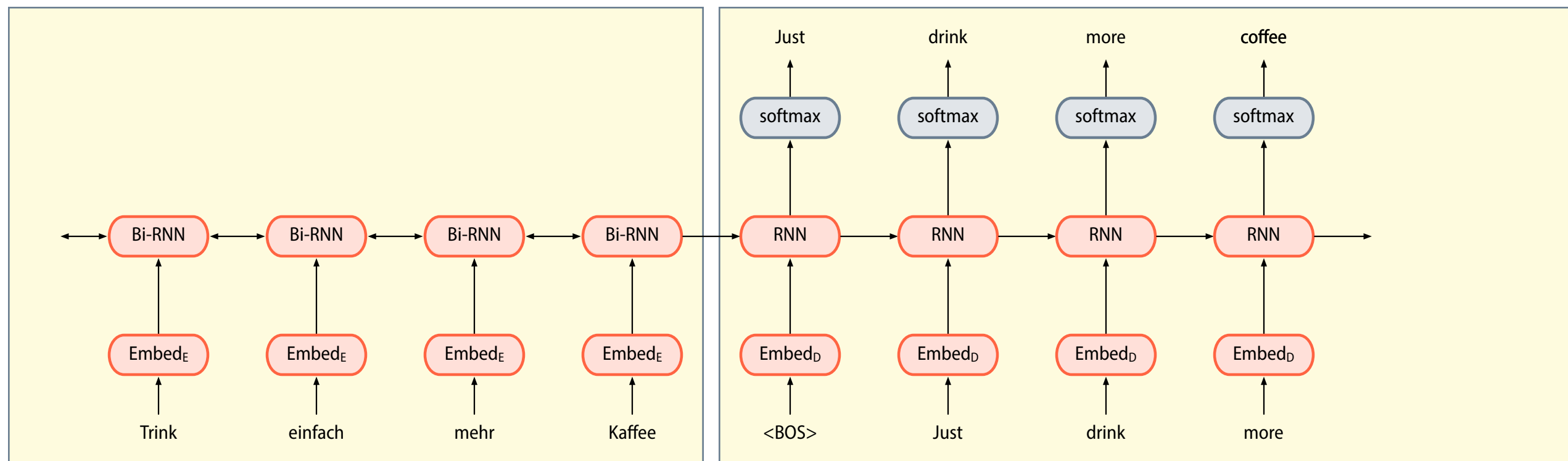


encoder

decoder

[Sutskever et al. \(2014\)](#)

Standard seq2seq-arkitekturen

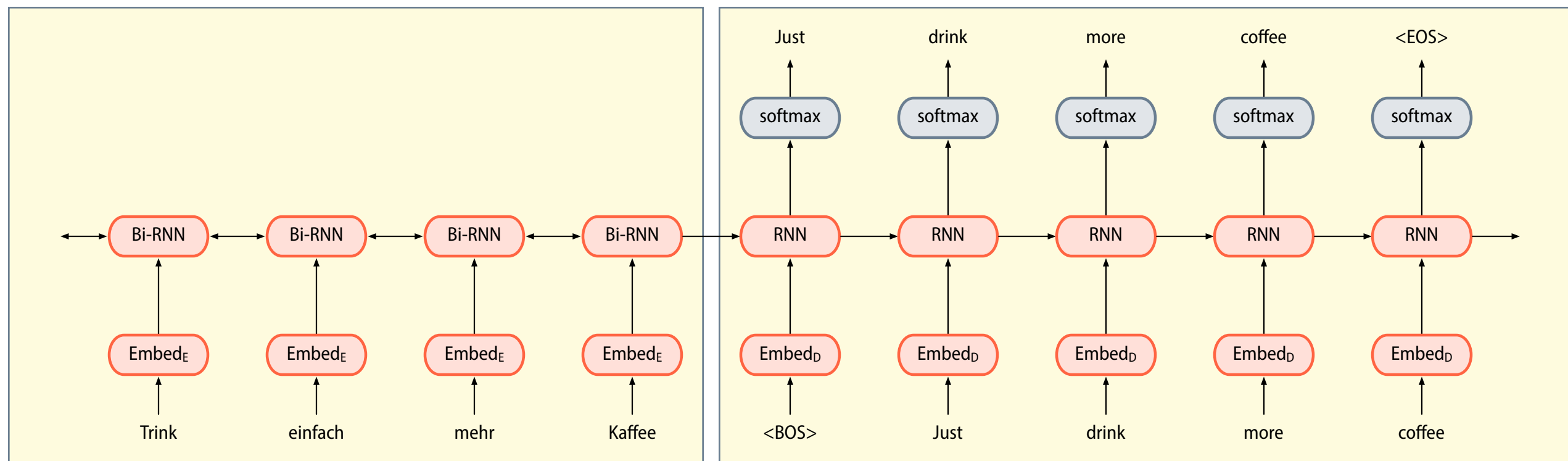


encoder

decoder

[Sutskever et al. \(2014\)](#)

Standard seq2seq-arkitekturen



encoder

decoder

[Sutskever et al. \(2014\)](#)

Egenskaper hos seq2seq-modellen

- Modellen kan tränas med hjälp av backpropagation. Det enda som krävs för detta är meningspar (källspråk–målspråk).
- Samma modell kan användas för en rad andra uppgifter, såsom textsammanfattning eller dialog.