Overview of Natural Language Processing Part II & ACS L390

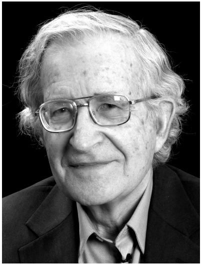
Lecture 9: Projection, Dependency and Attention

Weiwei Sun

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2024/25

I think the deepest property of language and puzzling property that's been discovered is what is sometimes called structure dependence. [...] Linear closeness is an easy computation, but here you're doing a much more, what looks like a more complex computation.

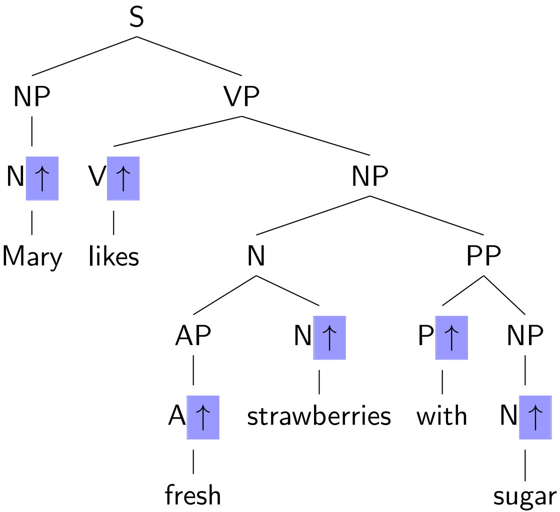


Lecture 9: Projection, Dependency and Attention

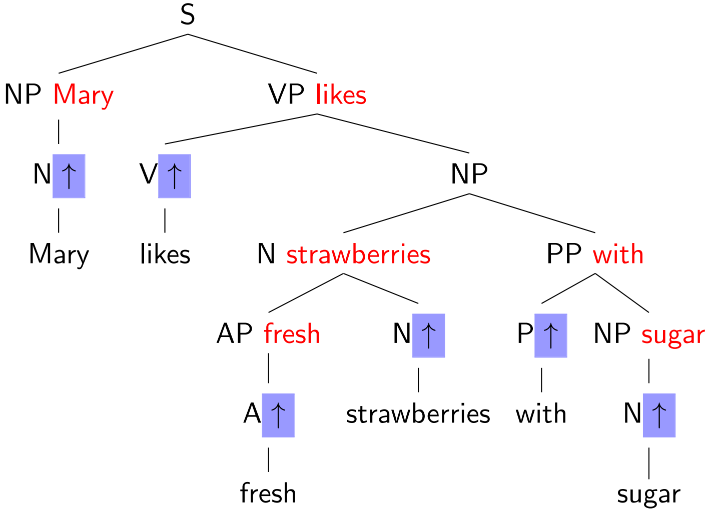
1. Projection and dependency
2. Mild context-sensitivity
3. Attention and transformer

Projection and dependency

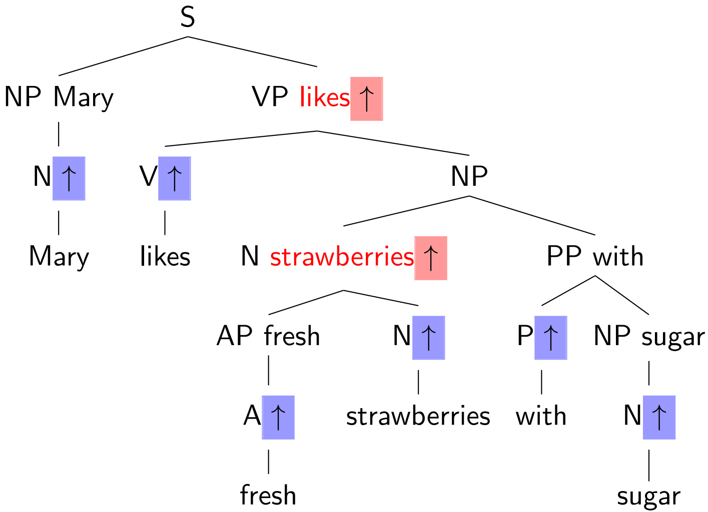
Projection and bi-lexical dependencies



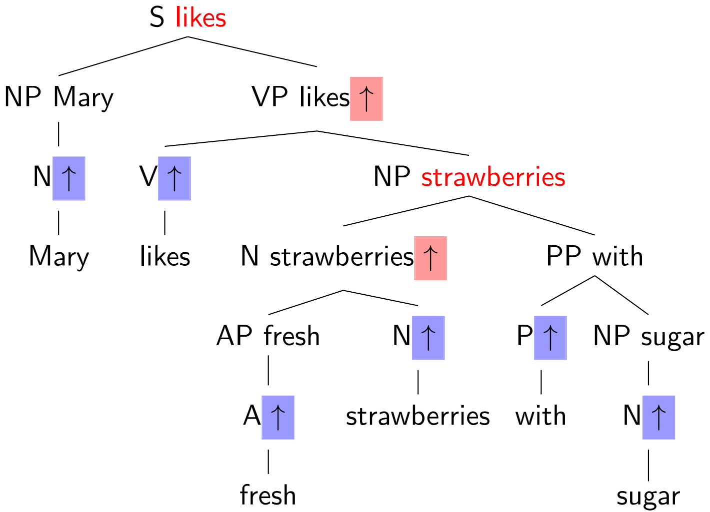
Projection and bi-lexical dependencies



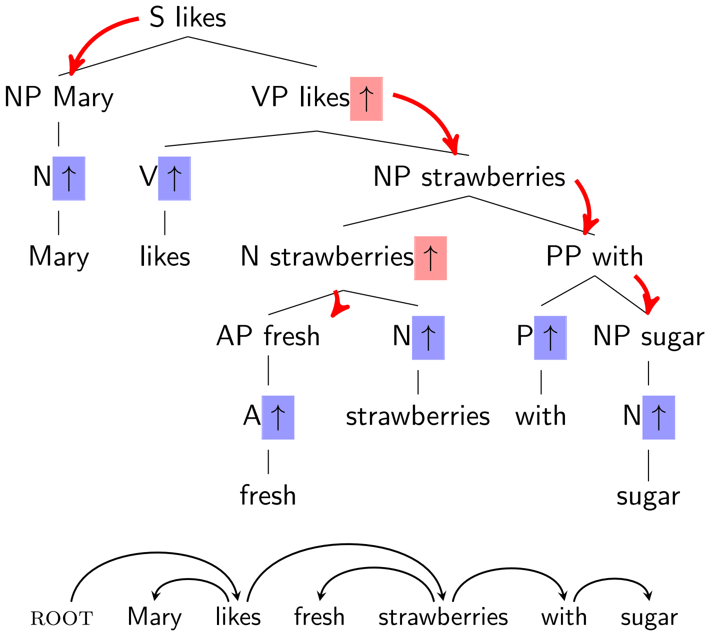
Projection and bi-lexical dependencies



Projection and bi-lexical dependencies



Projection and bi-lexical dependencies

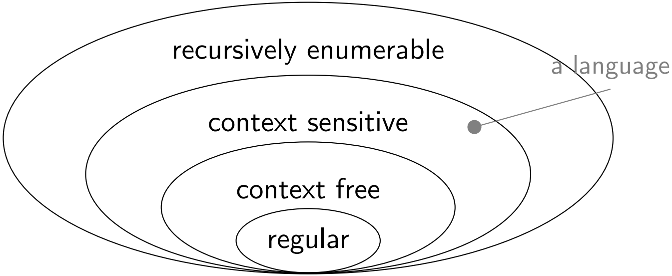


Mildly Context-Sensitive Languages

Reminder: Chomsky Hierarchy

Grammar	Languages	Production rules
Type-0	Recursively enumerable	$\alpha \to \gamma$
Type-1	Context-sensitive	$\alpha A \beta \to \alpha \gamma \beta$
Type-2	Context-free	$A \to \gamma$
Type-3	Regular	$A \to a$
		$A \to aB$

$$a \in N;\ \alpha, \beta \in (N \cup \Sigma)^*, \gamma \in (N \cup \Sigma)^+$$



Challenge

Cross-serial dependencies in Swiss German

. . . das mer em Hans es huus hälfed aastriiche
. . . that we Hans$_{Dat}$ house$_{Acc}$ help paint
. . . that we helped Hans paint the house

. . . das mer d'chind em Hans es huus lönd hälfe aastriiche
. . . that we the children$_{Acc}$ Hans$_{Dat}$ house$_{Acc}$ let help paint
. . . that we let the children help Hans paint the house

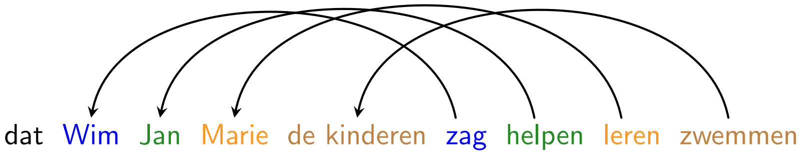
Cross-serial dependencies in Dutch

. . . dat Wim Jan Marie de kinderen zag helpen leren zwemmen
. . . that Wim Jan Marie the children saw help teach swim
. . . that Wim saw Jan help Marie teach the children to swim

Cross-serial dependencies

Cross-serial dependencies in Dutch

. . . dat Wim Jan Marie de kinderen zag helpen leren zwemmen
. . . that Wim Jan Marie the children saw help teach swim
. . . that Wim saw Jan help Marie teach the children to swim

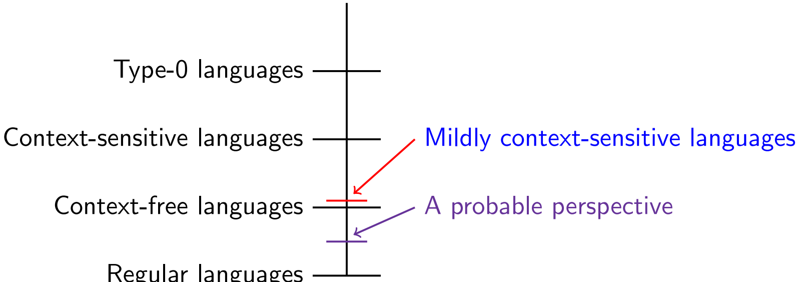


Mildly Context-Sensitive Languages

With a *possibility* perspective

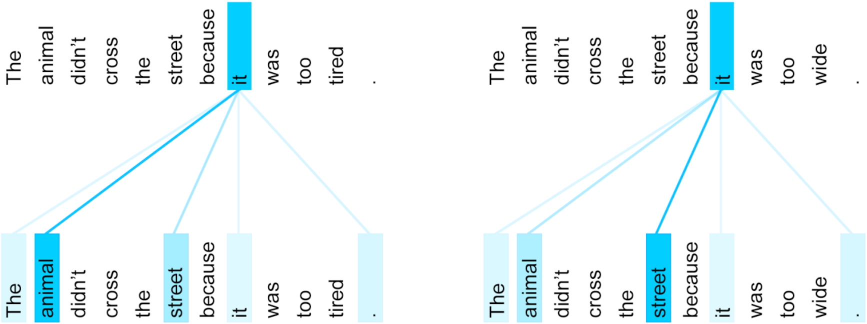
Natural languages are provably non-context-free.

Natural languages = mildly context-sensitive languages?



Coreference

(1) a. The **chicken** didn't cross the street because **it** was too tired.
b. The chicken didn't cross the **street** because **it** was too wide.



Attention and Transformer

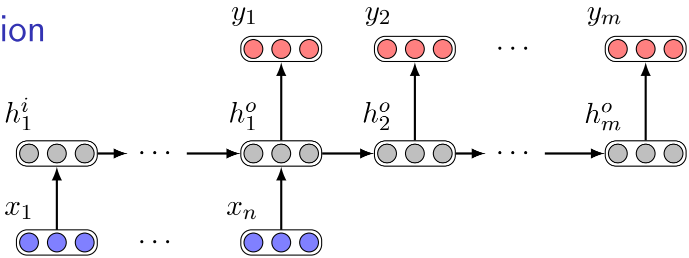
Key milestones

- Before 2014: RNN/LSTM
- Attention 2015: Bahdanau, Cho and Bengio. Neural machine translation by joint learning to align and translate.
- Transformer 2017: Google's "Attention is All Your Need"
- Pre-trained Models 2018–: BERT, GPT, etc.

Key milestones

- Before 2014: RNN/LSTM
- **Attention 2015: Bahdanau, Cho and Bengio. Neural machine translation by joint learning to align and translate.**
- Transformer 2017: Google's "Attention is All Your Need"
- Pre-trained Models 2018–: BERT, GPT, etc.

Attention



- We have encoder hidden states $h_i^i, \ldots, h_n^i \in \mathbb{R}^h$
- On timestep t, we have decoder hidden state $h_t^o \in \mathbb{R}^h$
- We get attention scores e^t for step t:

$$e^t = [(h_t^o)^\top h_1^i, (h_t^o)^\top h_2^i, \ldots, (h_t^o)^\top h_n^i] \in \mathbb{R}^n$$

- We get a distribution by applying `softmax`:

$$\alpha^t = \texttt{softmax}(e^t)$$

- A weighted sum of encoder hidden states is then derived:

$$\boldsymbol{a}^t = \sum_{k=1}^{n} \alpha_k h_k^i \in \mathbb{R}^h$$

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

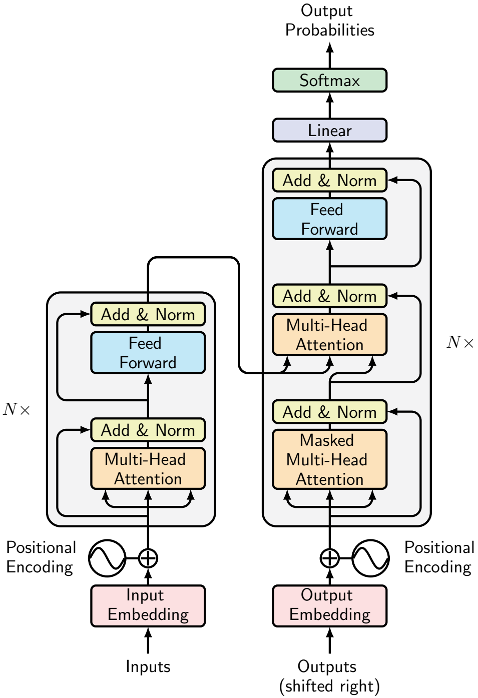
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Transformer

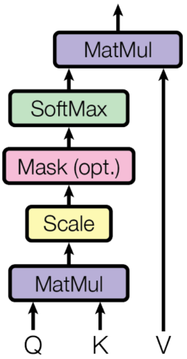
Transformer is all about (self) attention.

Self-attention is encoder-encoder (or decoder-decoder) attention where each word attends to each other word within the input (or output).

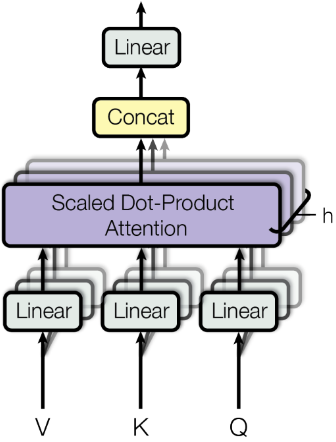


Multi-head attention

Scaled Dot-Product Attention



Multi-Head Attention



Self-attention

- For each word x_i, calculate its `query`, `key` and `value`:

$$q_i = W^Q x_i; \quad k_i = W^K x_i; \quad v_i = W^V x_i$$

- Calculate attention score between query and keys:

$$e_{ij} = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$

- Apply `softmax` to normalise attention scores:

$$\alpha_{ij} = \texttt{softmax}(e_{ij})$$

- Take a weighted sum of values:

$$o_j = \sum_j \alpha_{ij} v_j$$

- All words attend to all words in previous layer.

As Matrix multiplication

- Packing the input embeddings for the N tokens of the input sequence into a single matrix: $X \in \mathbb{R}^{N \times d}$. Each row is the embedding of one token.
- Key and query matrices (denoted as W^K and W^Q) are of size $d \times d_k$
- Value matrix is of size $d \times d_v$

$$Q = XW^Q \qquad K = XW^K \qquad V = XW^V$$

Attention matrix

$$A = \texttt{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Multi-head attention

$$\boldsymbol{head}_i = \textsc{SelfAttention}(Q^i, K^i, V^i)$$

$$\textsc{MultiHeadAttention}(X) = \texttt{concat}(\boldsymbol{head}_1, \boldsymbol{head}_2, ...)W^O$$

Residual connection and layer normalisation

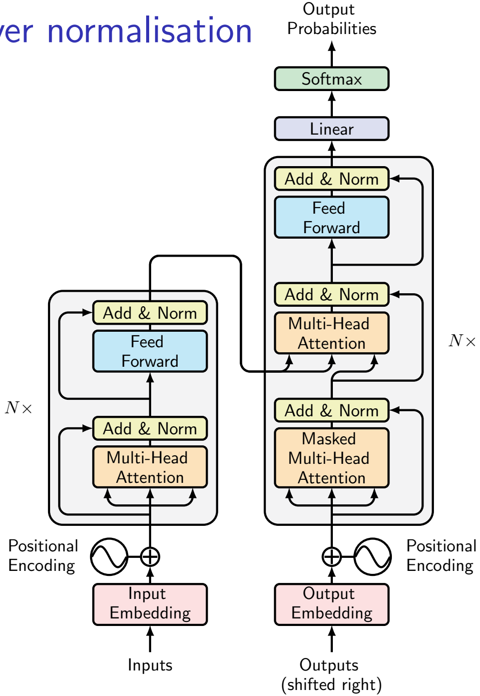
- Residual connection:
  $x \mapsto f(x) + x$.
- LayerNorm is applied to a single vector in a hidden layer.

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i$$

$$\sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2}$$

$$\hat{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i - \mu}{\sigma}$$

$$\textsc{LayerNorm}(\mathbf{x}) = \gamma\hat{\mathbf{x}} + \beta$$



Reading

D Jurafsky and J Martin. *Speech and Language Processing*.

- §18.1 and §18.4. Dependency Parsing. *Speech and Language Processing*. D Jurafsky and J Martin. `https://web.stanford.edu/~jurafsky/slp3/18.pdf`
- Chapter 9. The Transformer. `https://web.stanford.edu/~jurafsky/slp3/9.pdf`.