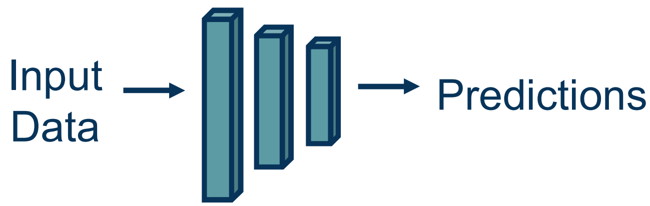
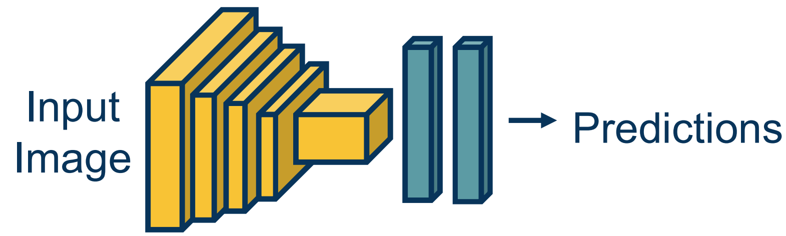
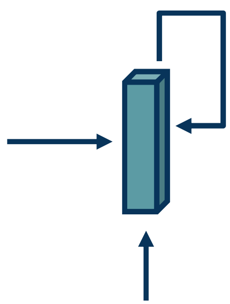
Module 3 Introduction



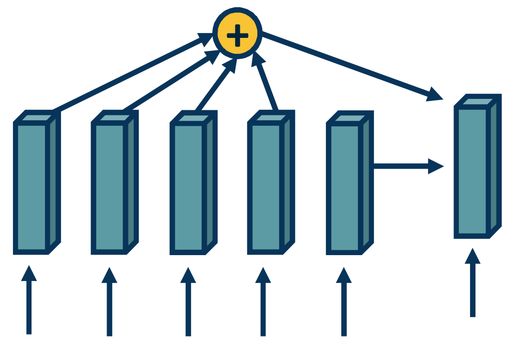
Fully Connected Neural Networks



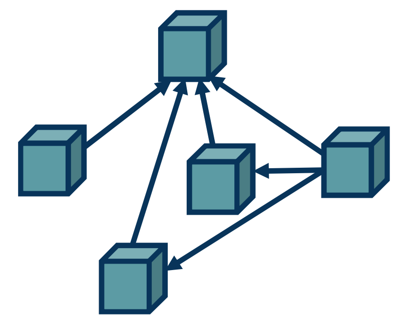
Convolutional Neural Networks



Recurrent Neural Networks

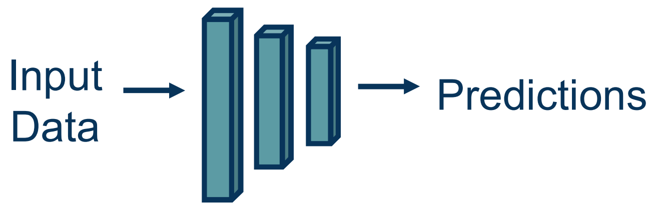


Attention-Based Networks

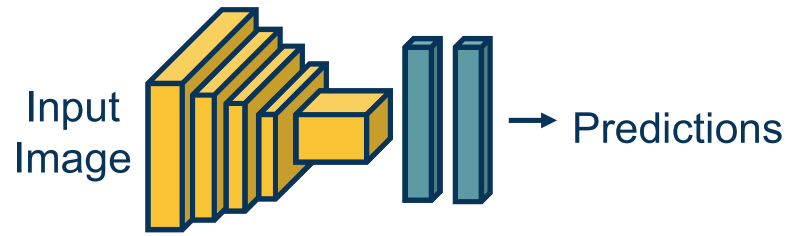


Graph-Based Networks

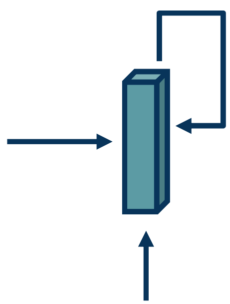
The Space of Architectures



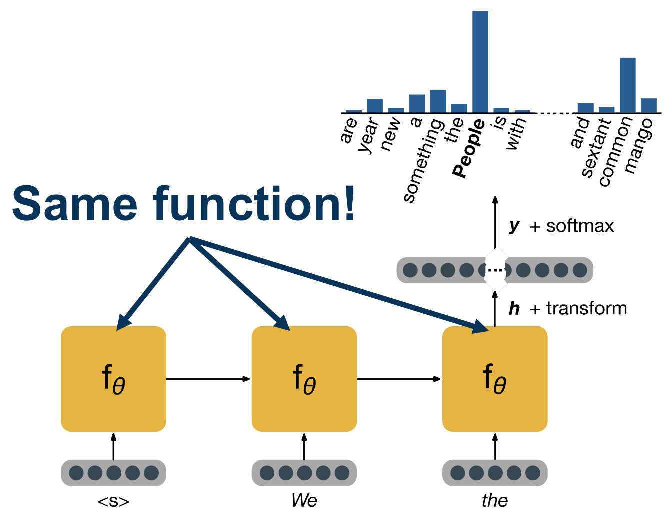
Fully Connected Neural Networks



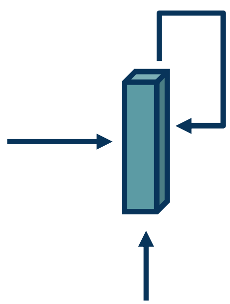
Convolutional Neural Networks



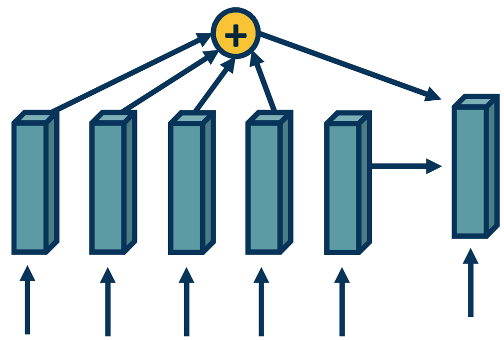
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks



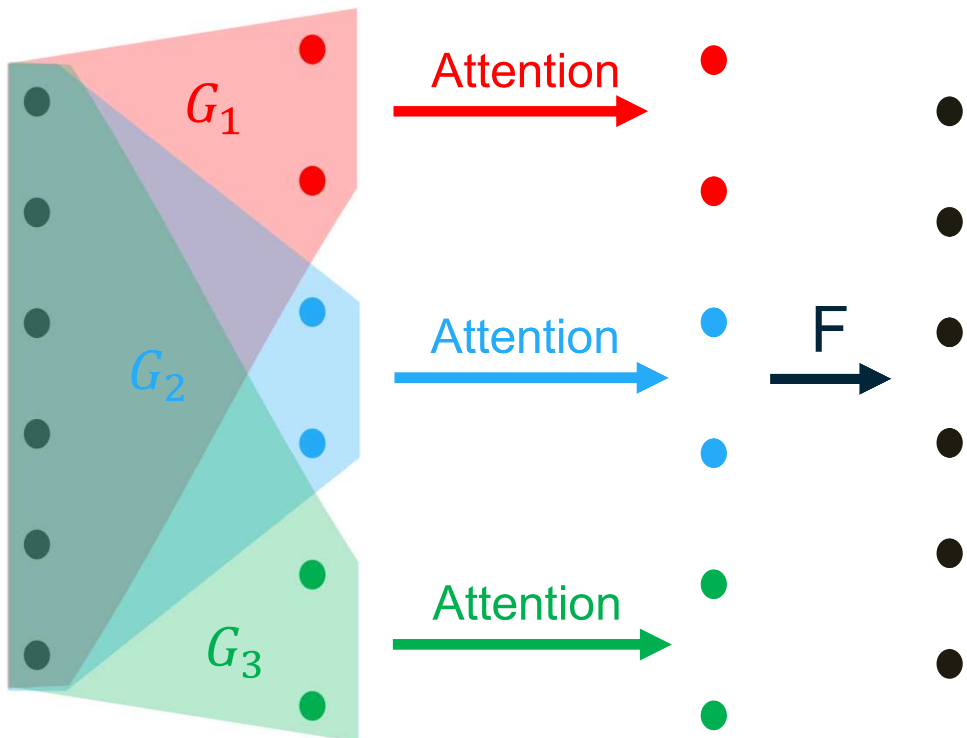
Attention-Based Networks

The Space of Architectures

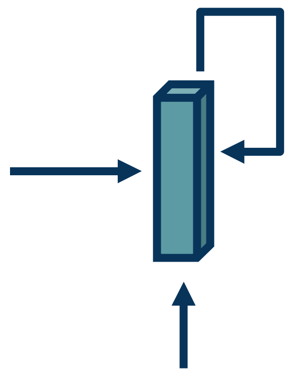
Transformer [Vaswani et. al. 2017] is a multi-layer attention model that is currently state of the art in most language tasks (and in many other things!)

Has superior performance compared to previous attention based architectures via

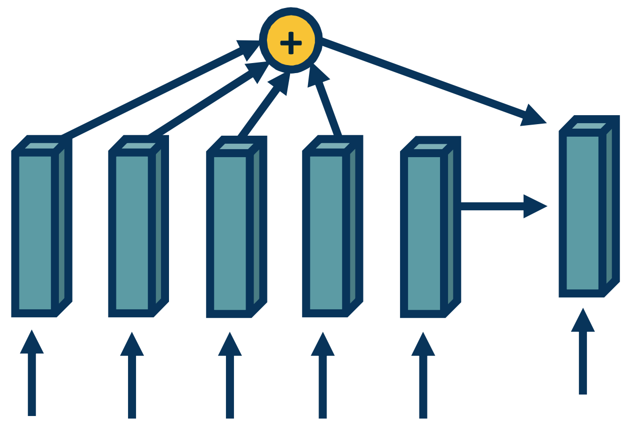
- Multi-query hidden-state propagation (“self-attention”)
- **Multi-head attention**
- Residual connections, LayerNorm



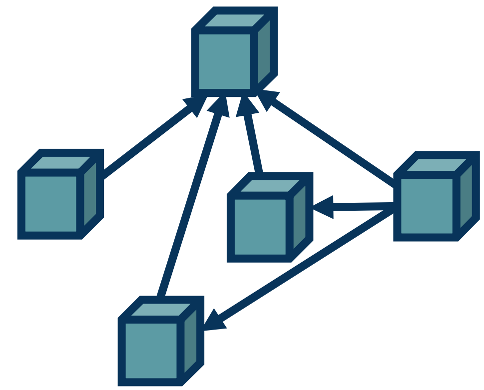
Enter the Transformer



Recurrent
Neural Networks



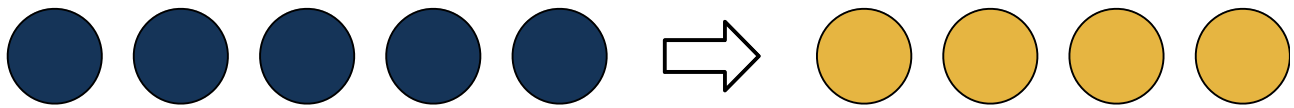
Attention-Based
Networks



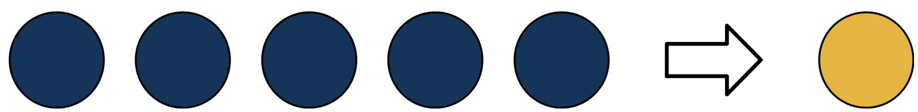
Graph-Based
Networks

The Space of Architectures

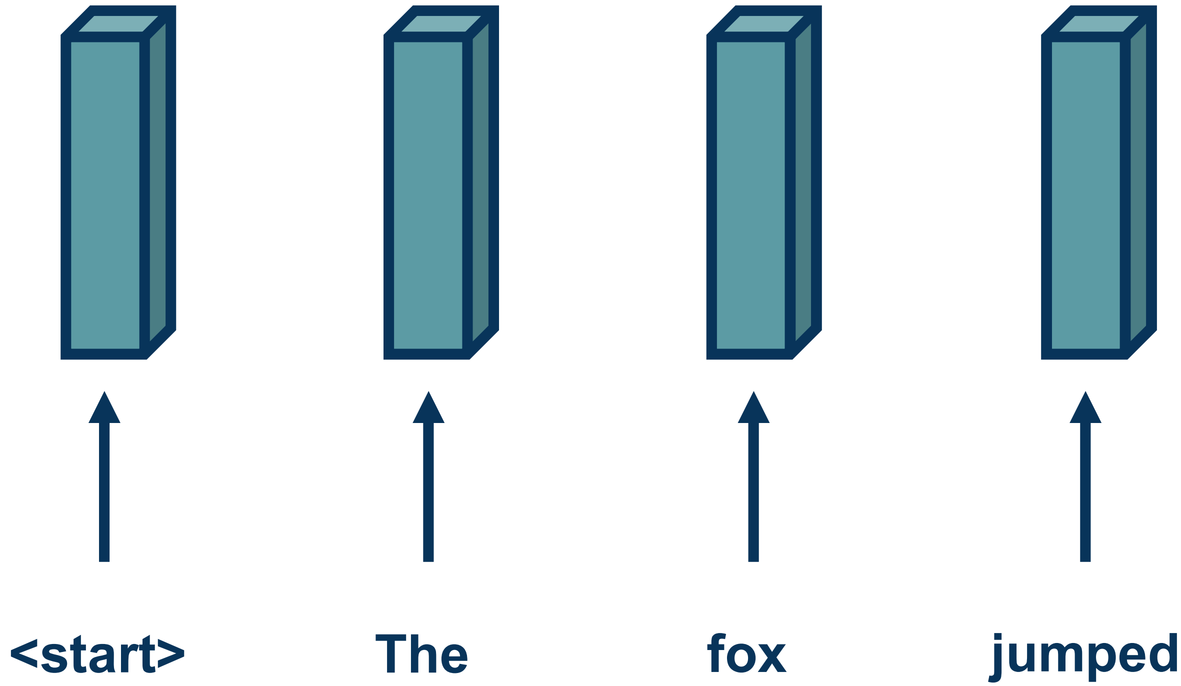
- **Many → many**: speech recognition, optical character recognition



- **Many → one**: sentiment analysis, topic classification



- Also consider: **one → many, one → one.**

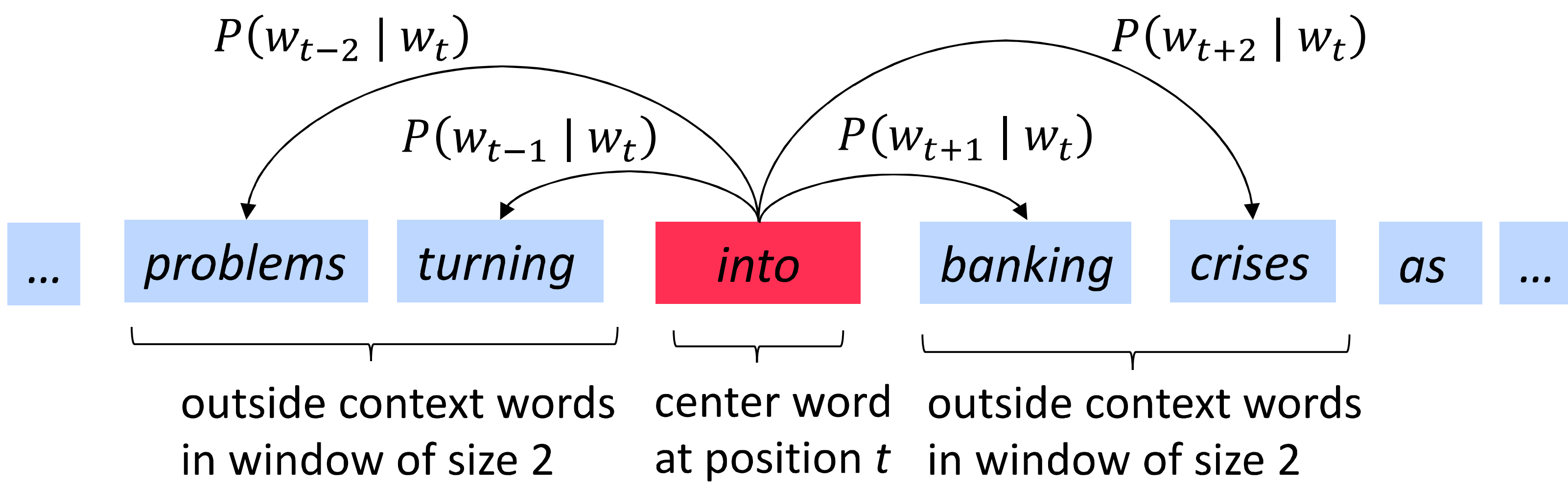


Example Application: NLP

$$
\begin{aligned}
p(\boldsymbol{s}) &= p(w_1, w_2, \ldots, w_n) \\
&= p(w_1)\, p(w_2 \mid w_1)\, p(w_3 \mid w_1, w_2) \cdots p(w_n \mid w_{n-1}, \ldots, w_1) \\
&= \prod_i p(\underbrace{w_i}_{\text{next word}} \mid \underbrace{w_{i-1}, \ldots, w_1}_{\text{history}})
\end{aligned}
$$

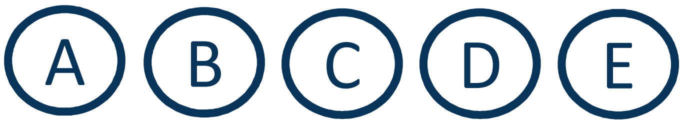
Word2vec: the Skip-gram model

- The idea: use words to **predict** their context words
- Context: a fixed window of size ***2m***

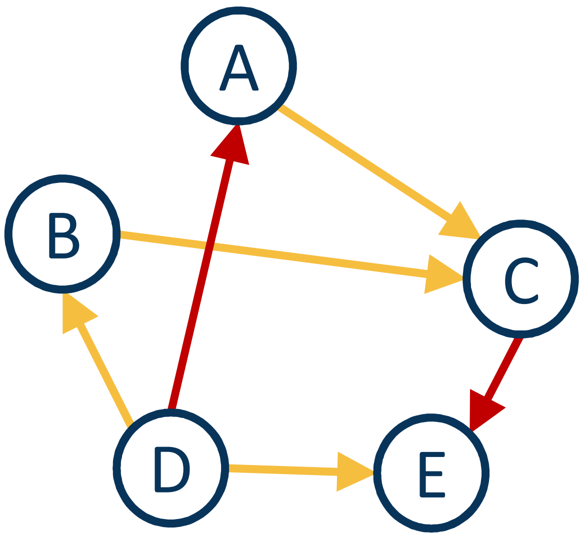


Slide Credit: Richard Socher, Christopher Manning

Graph Embeddings



A Sequential Structure



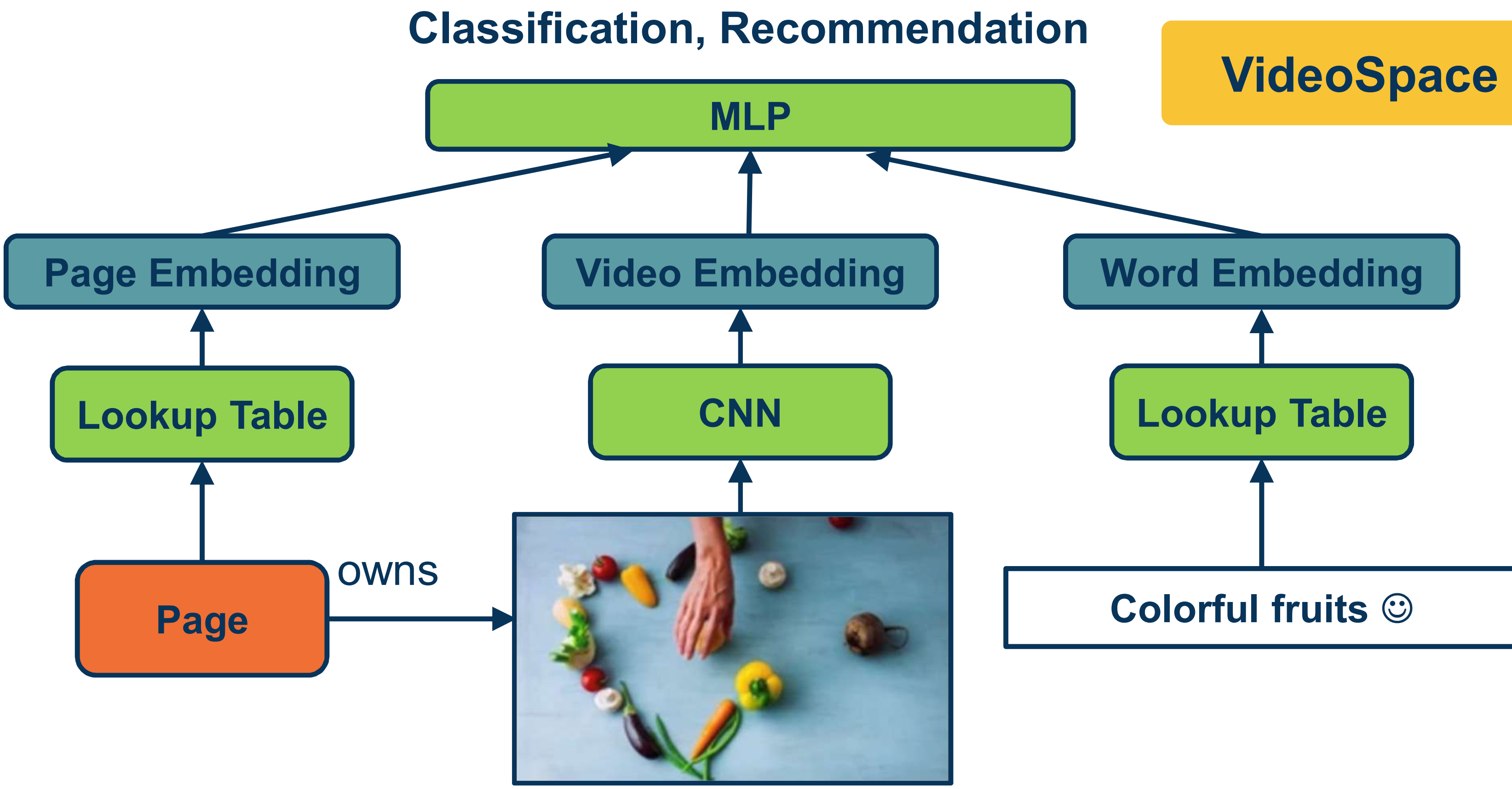
A Multi-Relation Graph

Embedding: A learned map from entities to vectors of numbers that encodes similarity

- Word embeddings: word → vector
- Graph embeddings: node → vector

Graph Embedding: Optimize the objective that **connected nodes have more similar embeddings** than unconnected nodes via gradient descent.

Slide Credit: Adam Lerer



VideoSpace

Application: VideoSpace

映画を見た → MT → I saw a movie

$$\boldsymbol{t} = argmax_{\,t}\, p(\boldsymbol{t}|\boldsymbol{s})$$

search

model

Alignment in machine translation: for each word in the target, get a distribution over words in the source [Brown et. al. 1993], (lots more)

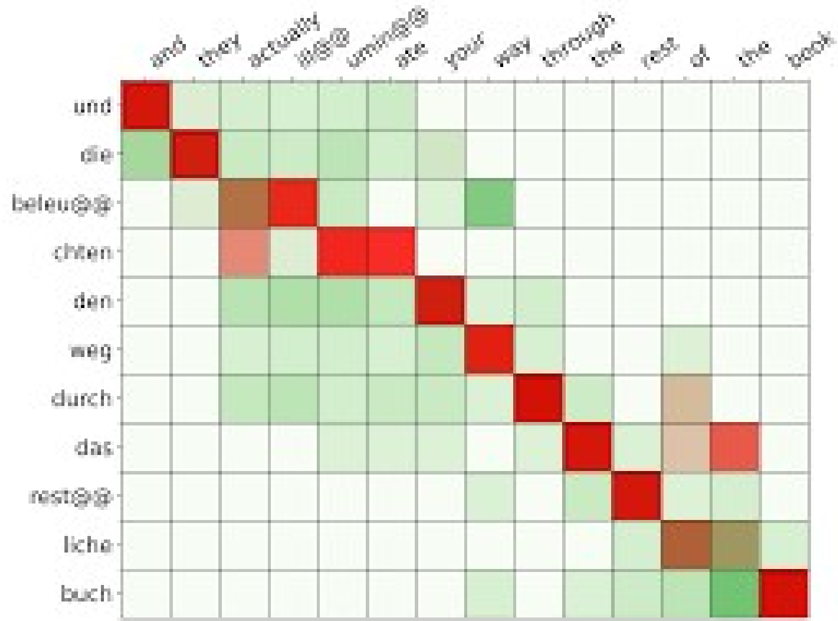
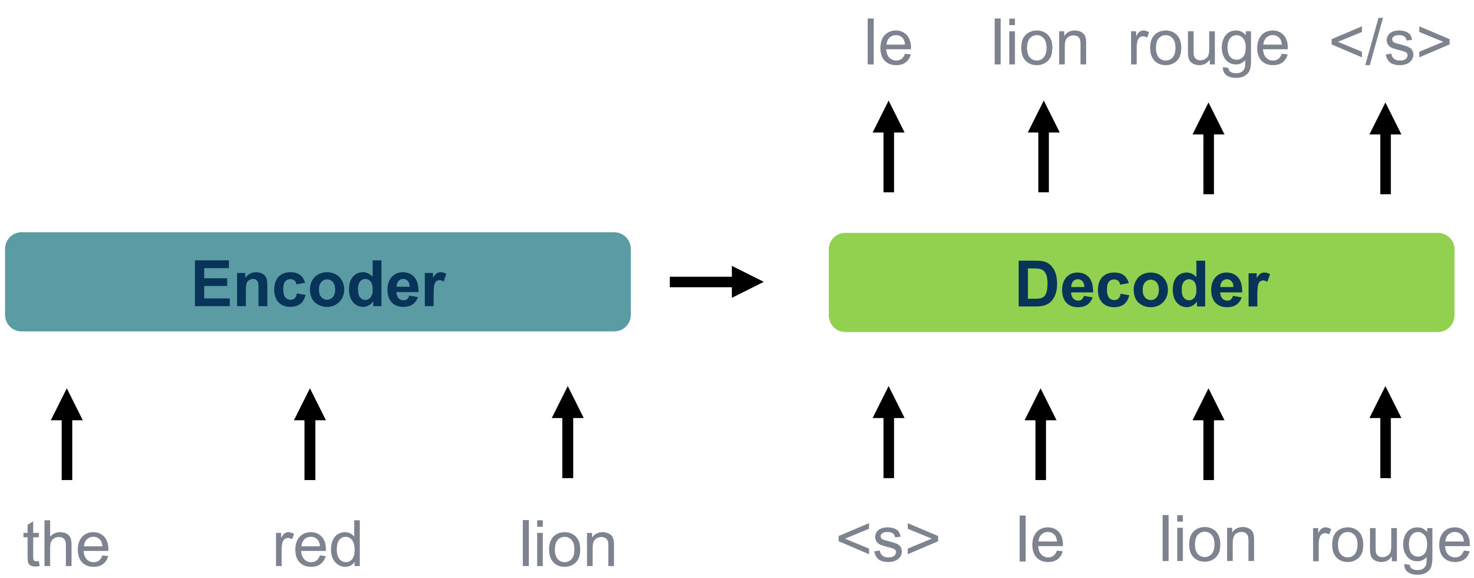
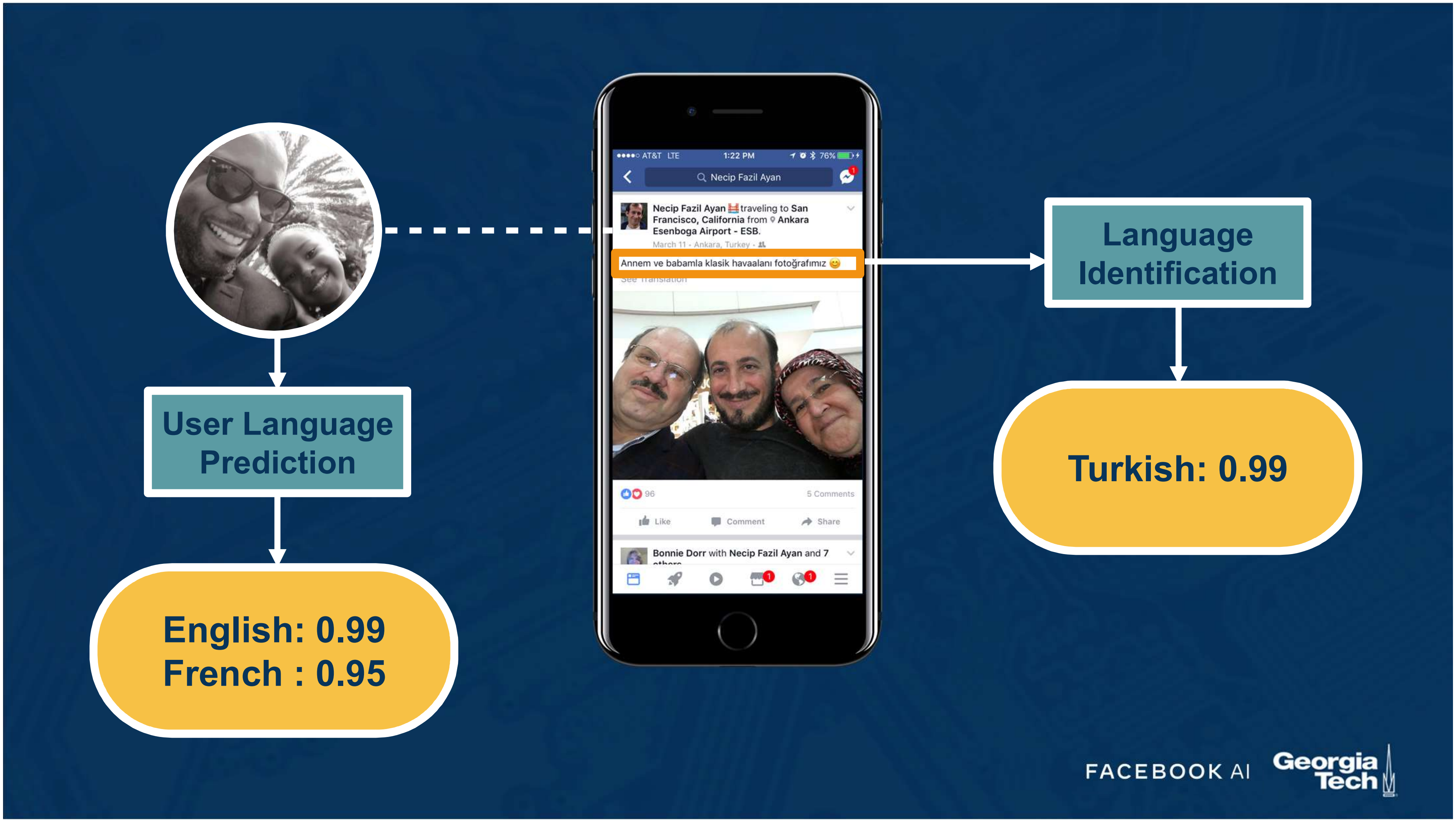
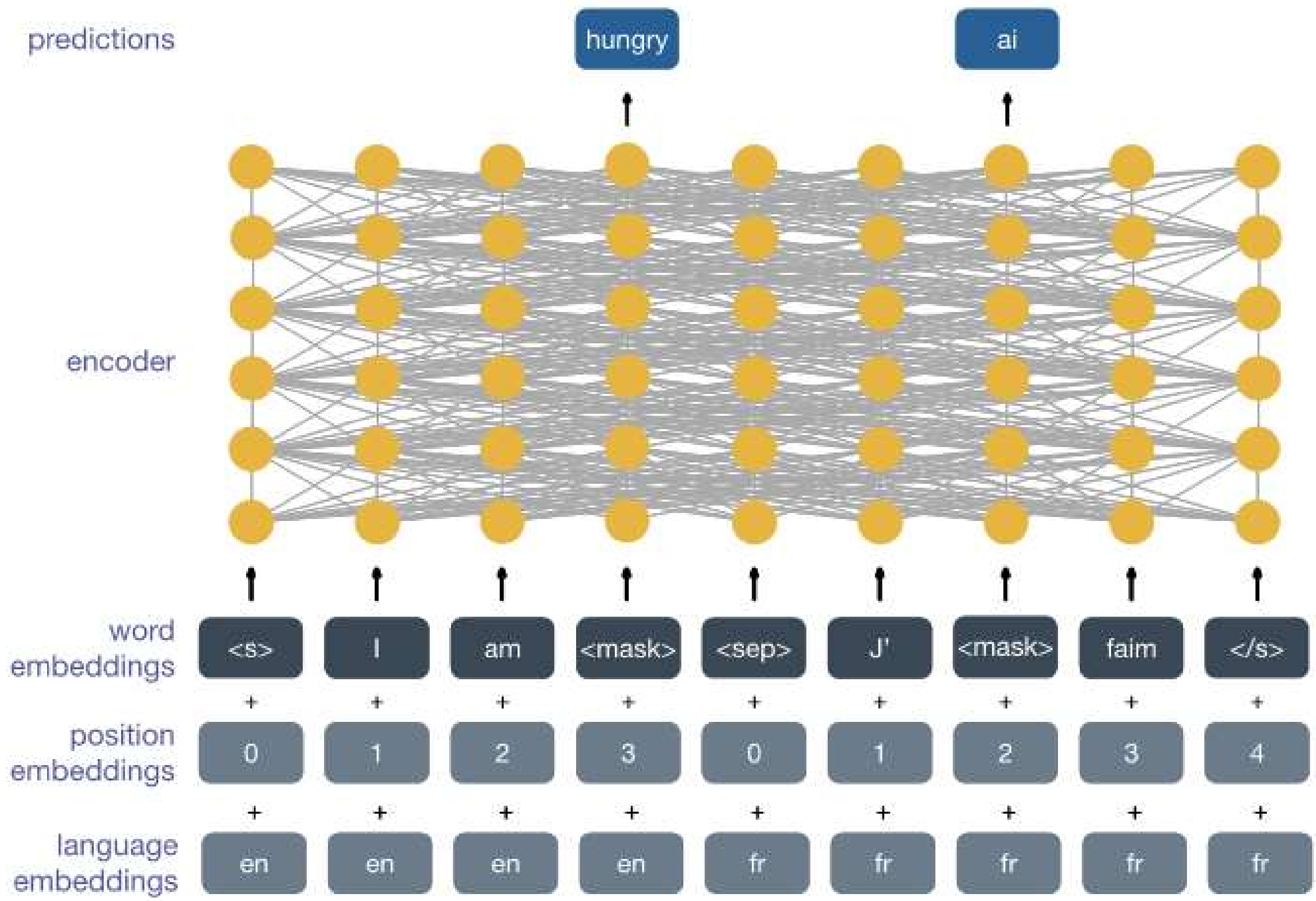


Figure from Latent Alignment and Variational Attention by Deng et. al.


le
lion
rouge
</s>
Encoder
Decoder
the
red
lion
<s>
le
lion
rouge



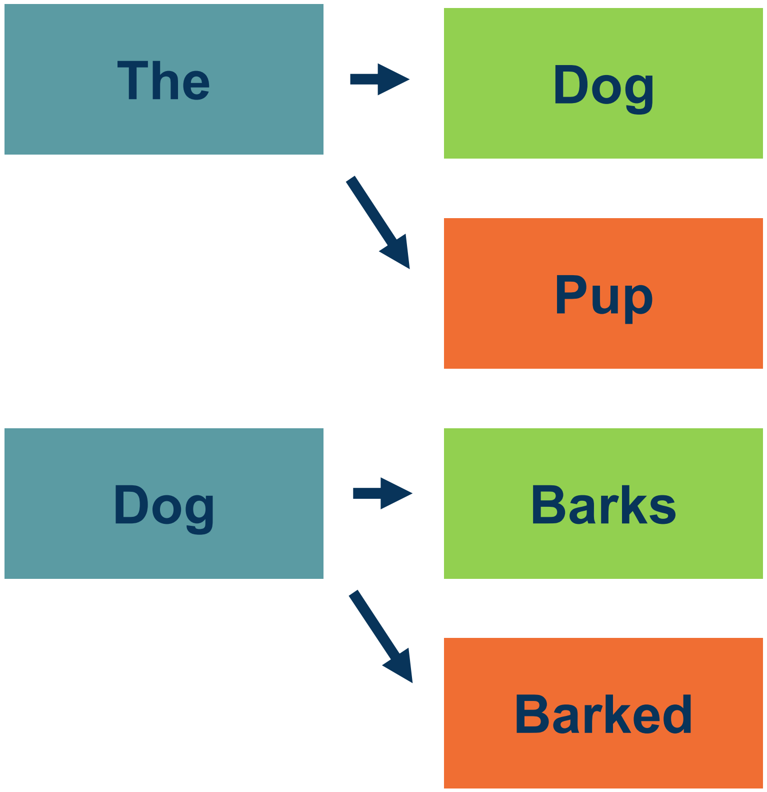
AT&T LTE
1:22 PM
76%
Necip Fazil Ayan
Necip Fazil Ayan traveling to San Francisco, California from Ankara Esenboga Airport - ESB.
March 11 · Ankara, Turkey
Annem ve babamla klasik havaalanı fotoğrafımız
96
5 Comments
Like
Comment
Share
Bonnie Dorr with Necip Fazil Ayan and 7
User Language Prediction
English: 0.99
French : 0.95
Language Identification
Turkish: 0.99
FACEBOOK AI
Georgia Tech



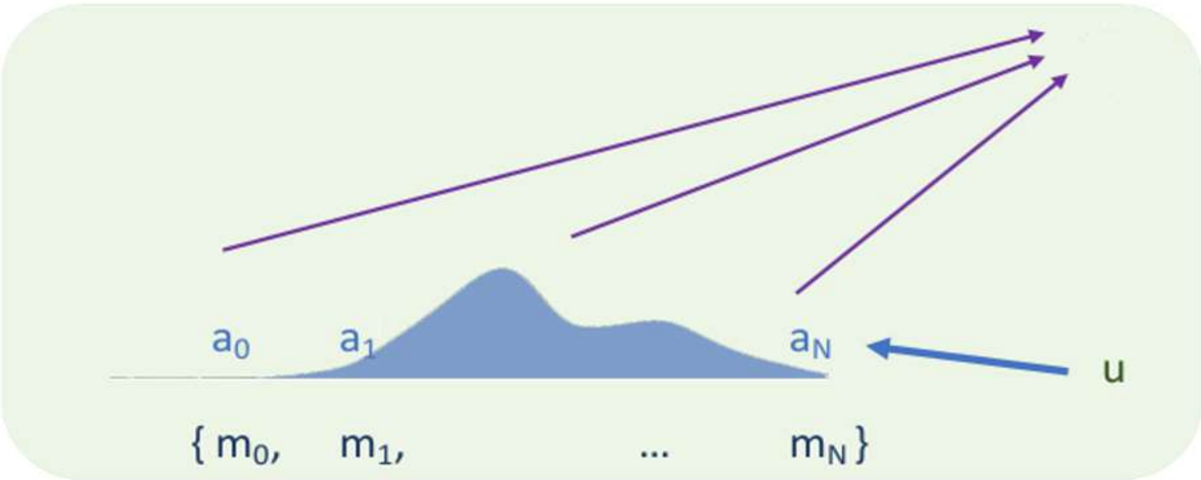
Cross-Lingual Masked Language Modeling

FACEBOOK AI Georgia Tech

- Search exponential space in linear time
- Beam size ***k*** determines "width" of search
- At each step, extend each of ***k*** elements by one token
- Top ***k*** overall then become the hypotheses for next step



- Given a set of vectors $\{\boldsymbol{u_1}, \ldots, \boldsymbol{u_N}\}$ and a "query" vector $\boldsymbol{q}$
- We can select the most similar vector to $\boldsymbol{q}$ via $\boldsymbol{p} = \boldsymbol{Softmax}(\boldsymbol{Uq})$



$$\boldsymbol{a_j} = \frac{\boldsymbol{e}^{\boldsymbol{u_j} \cdot \boldsymbol{q}}}{\sum_k \boldsymbol{e}^{\boldsymbol{u_k} \cdot \boldsymbol{q}}} \qquad \textbf{output} = \sum_k \boldsymbol{a_k u_k}$$