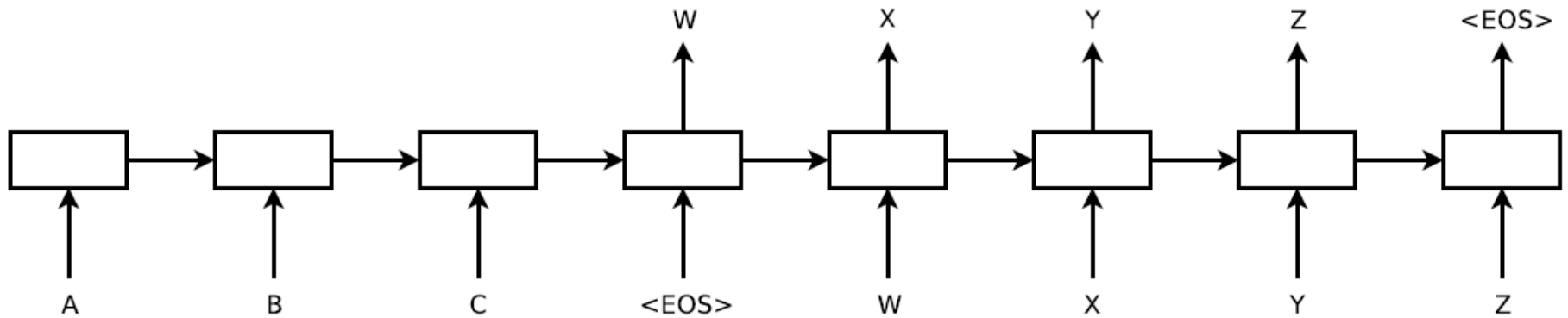
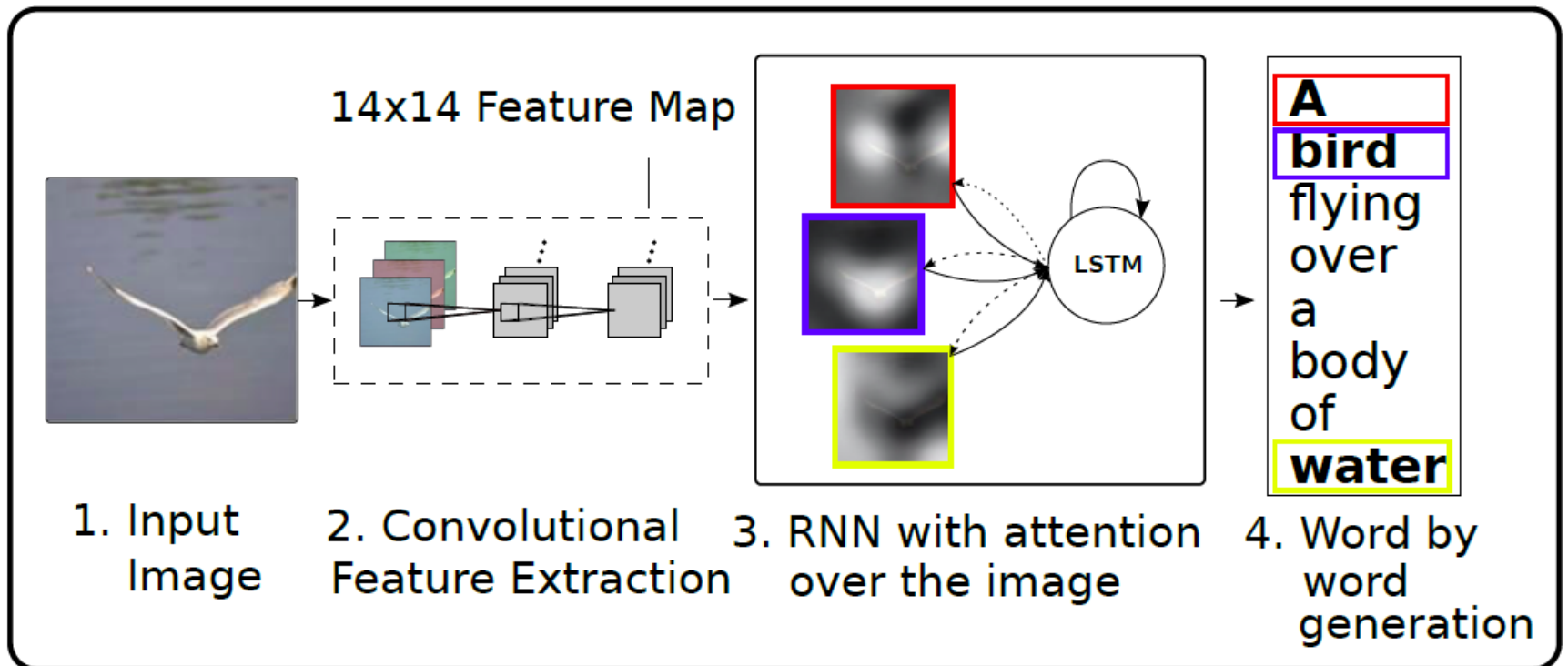


RNNs are cool

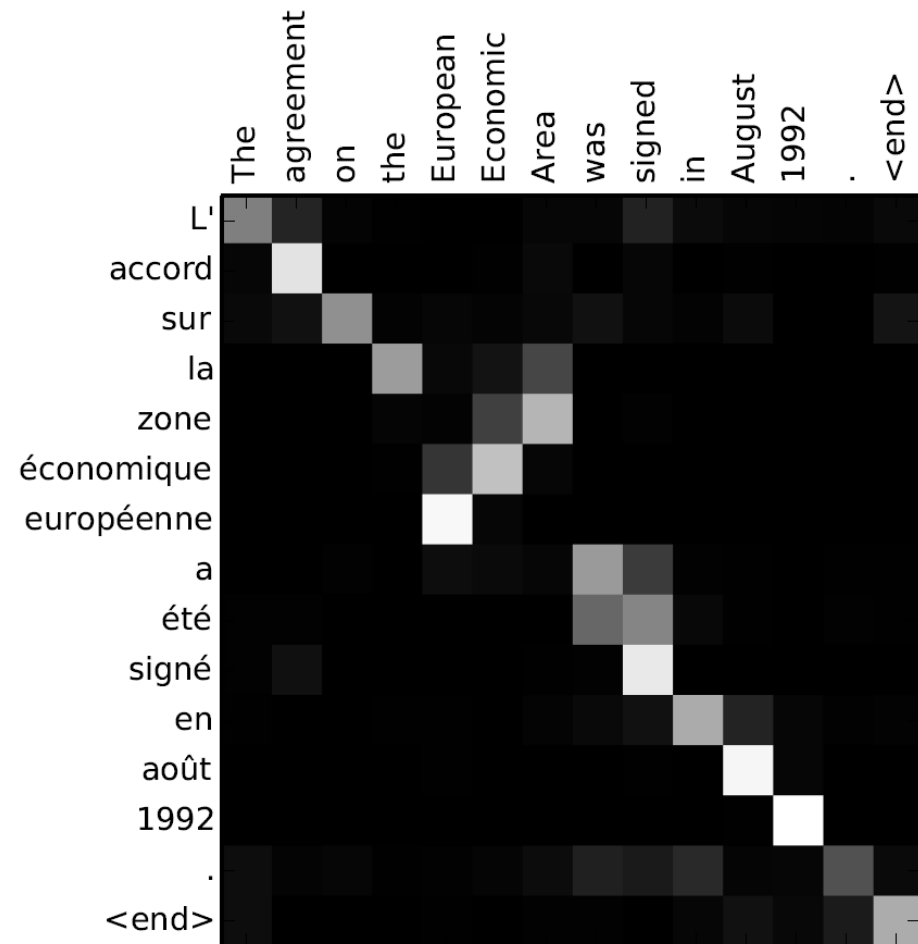
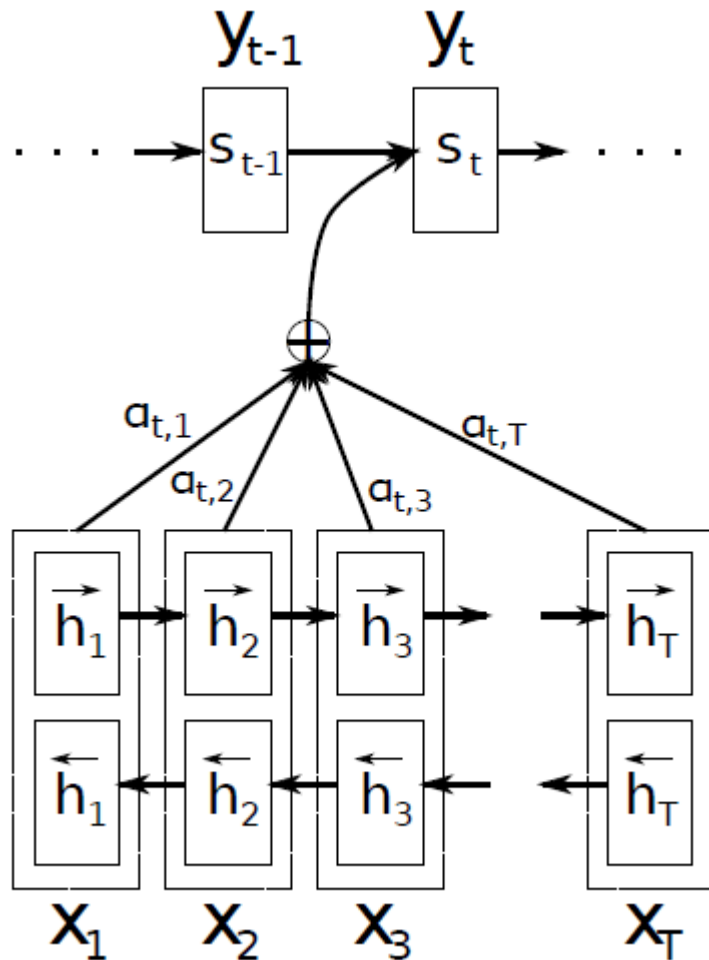
# Translation



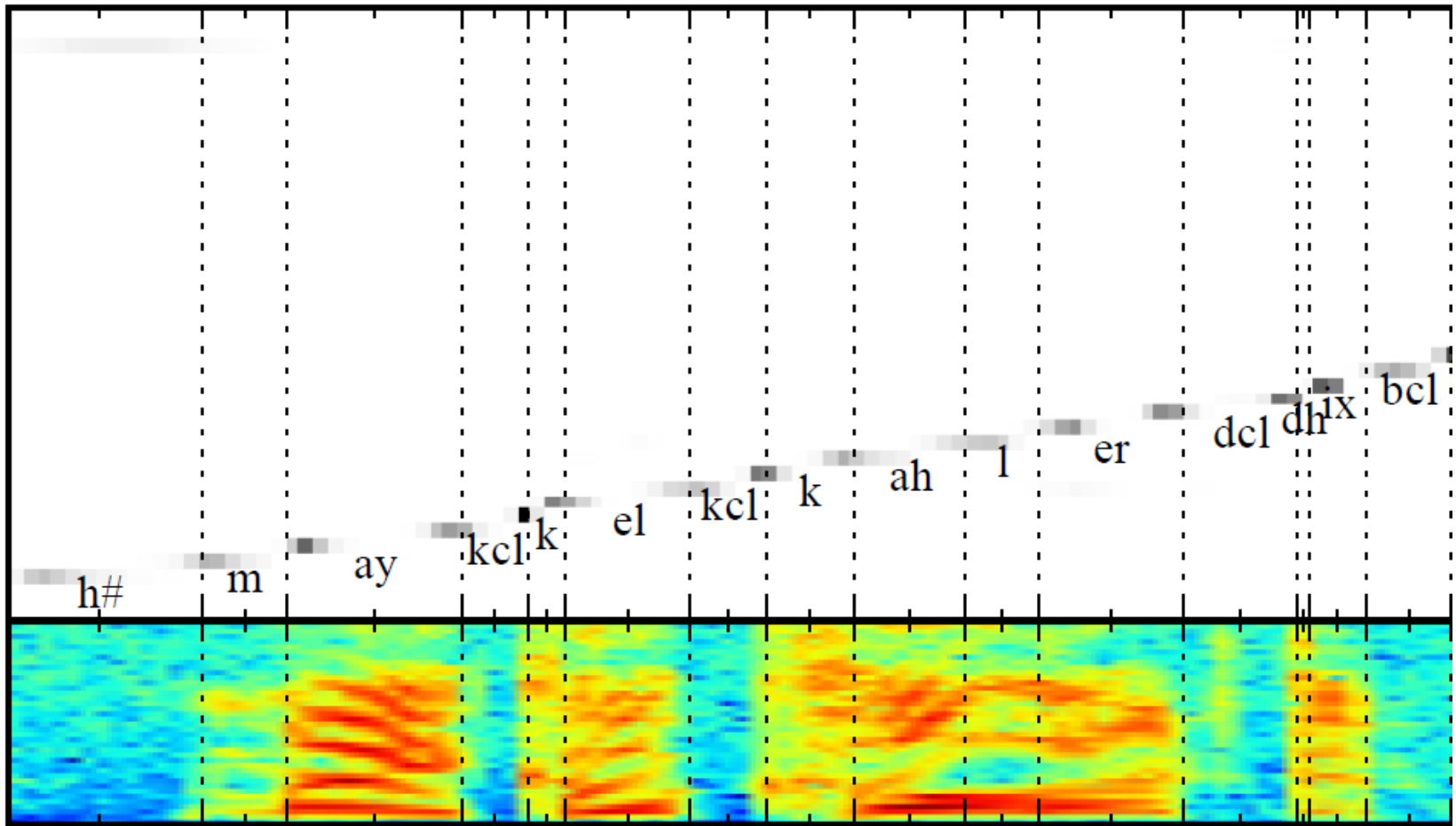
# Captions



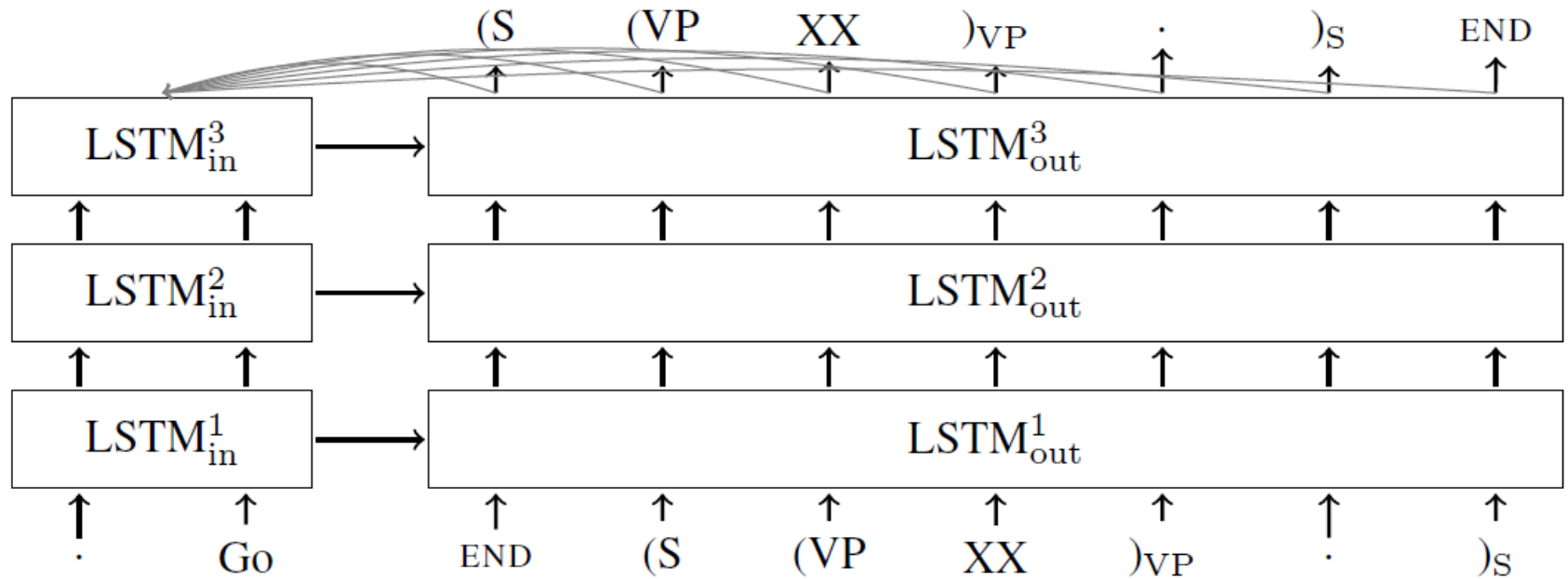
# Translation with attention



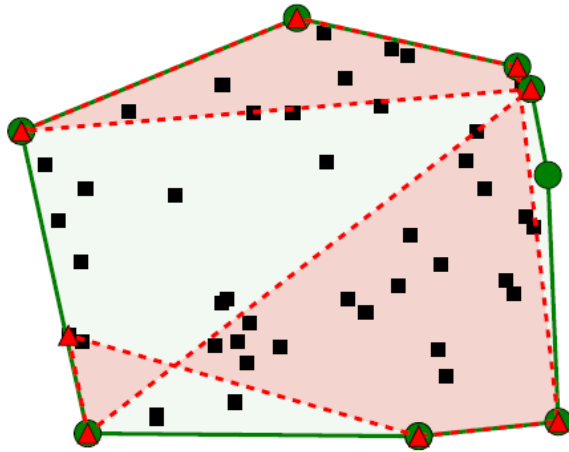
# Speech recognition



# Parsing

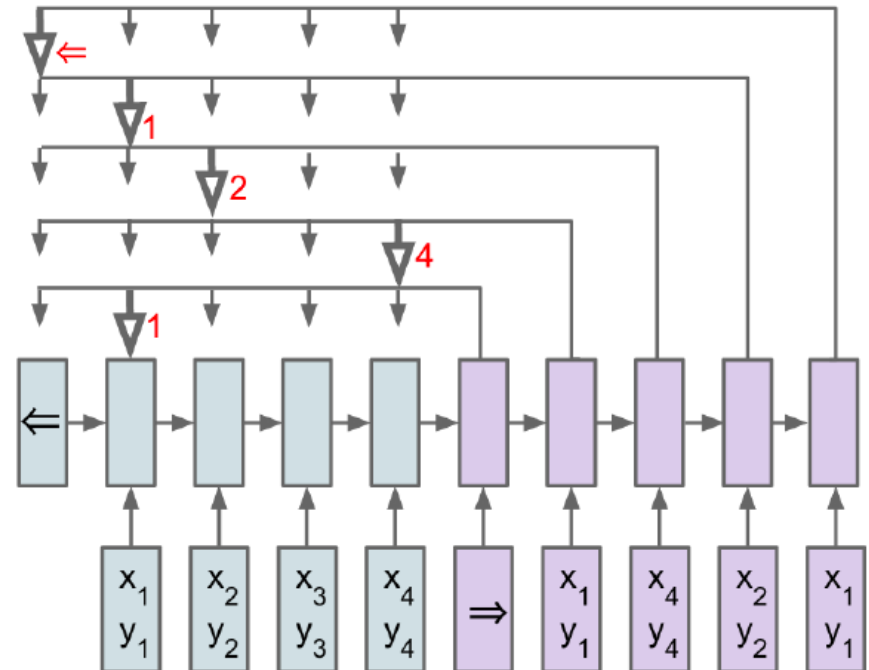
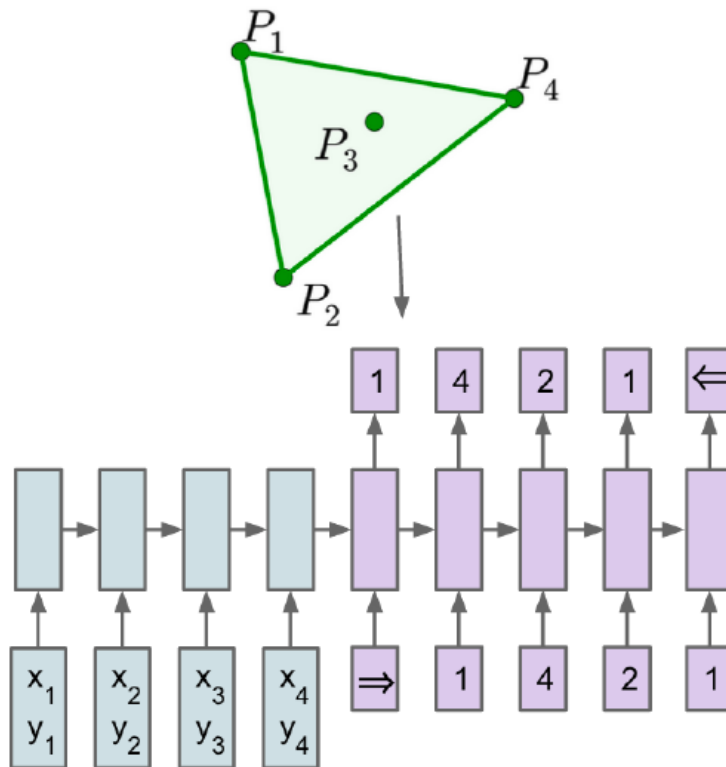


● Ground Truth    ▲ Predictions

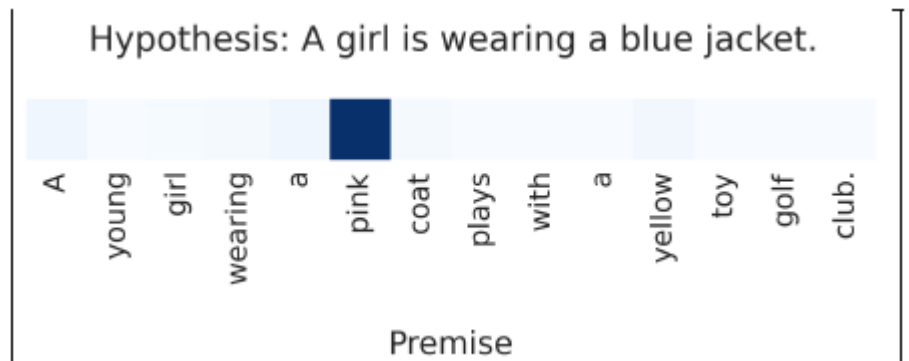
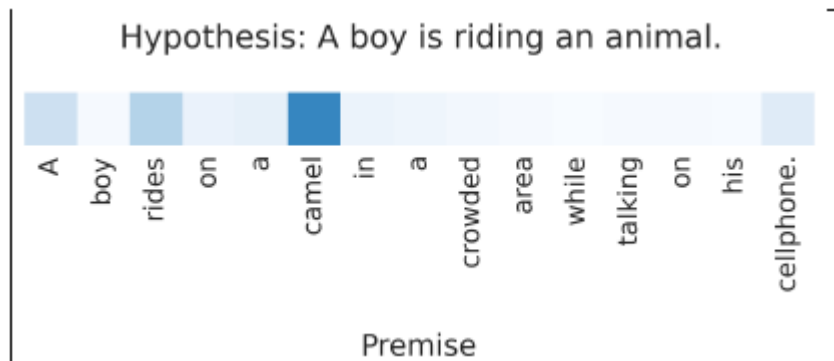
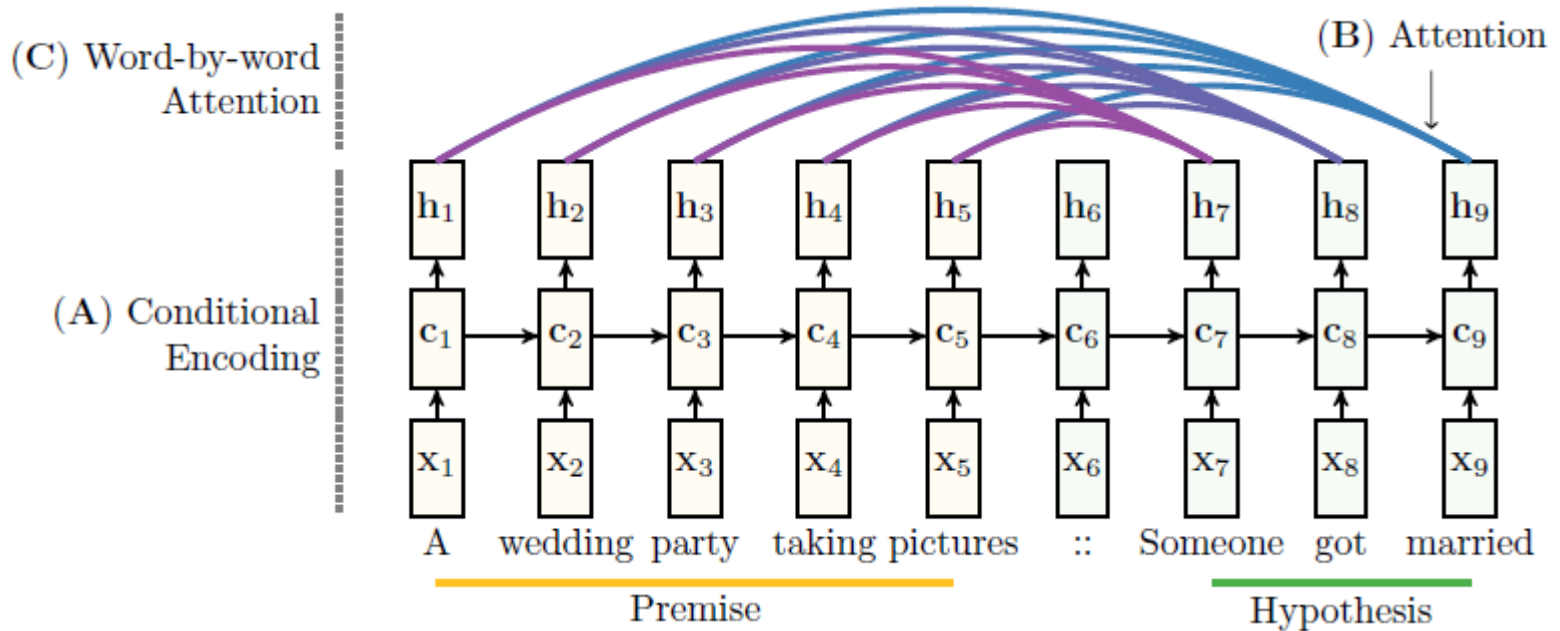


# Convex Hulls & TSP

<http://papers.nips.cc/paper/5866-pointer-networks.pdf>

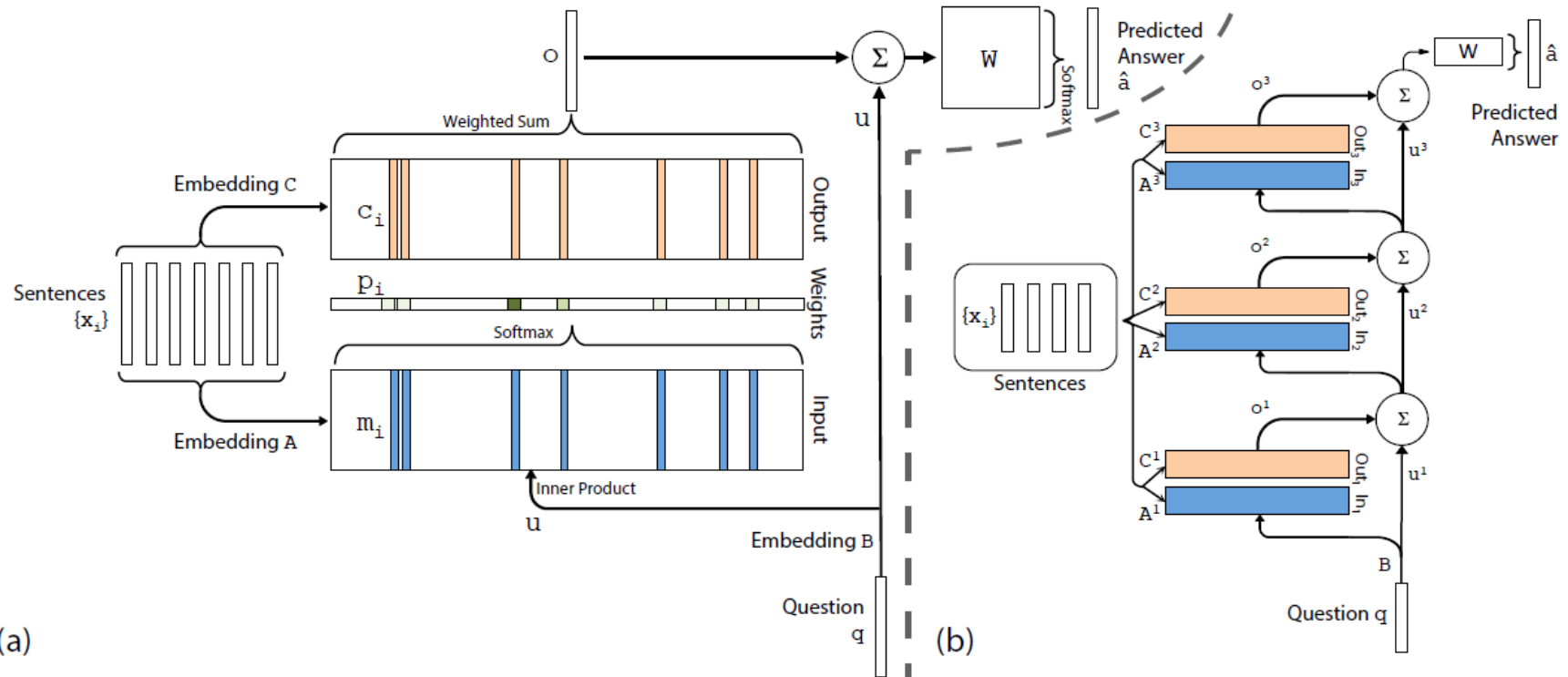


# Reasoning





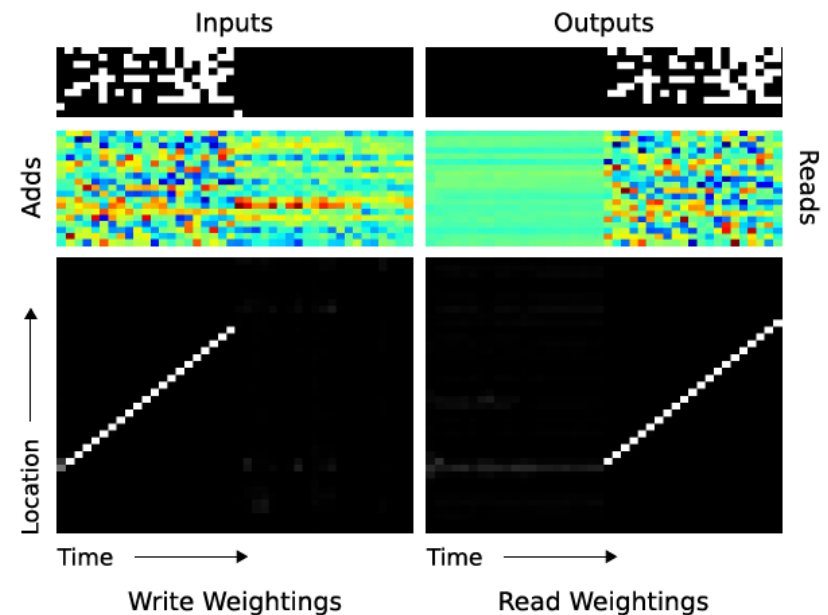
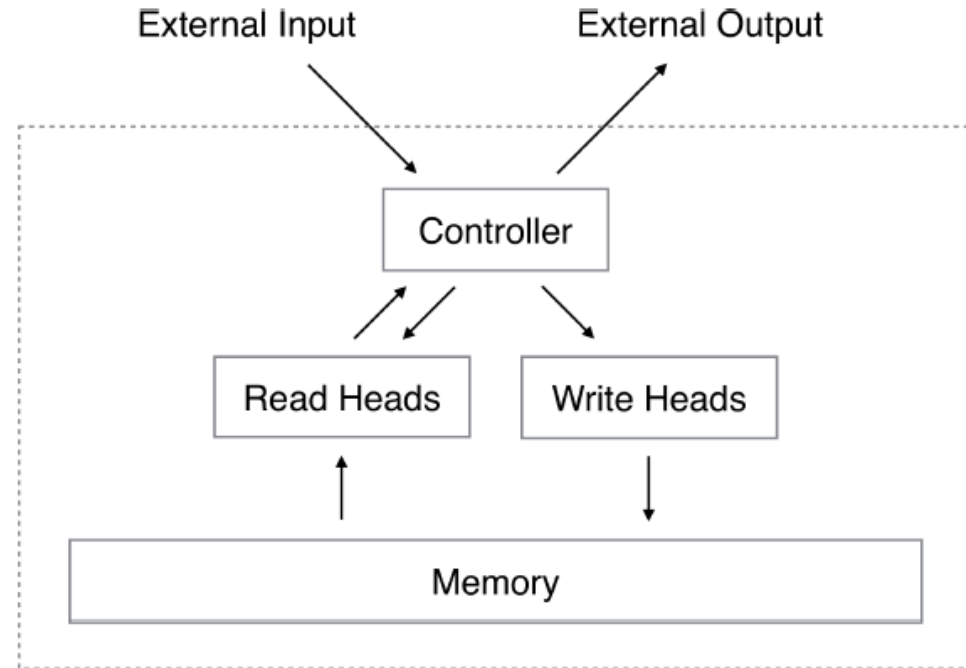
# Reasoning – facts in memory



Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
<b>What color is Greg? Answer: yellow Prediction: yellow</b>				

# Neural Turing Machine

- Typical RNN accesses and modifies all hidden state
- Use attention to read part of memory and to write to part of memory
- The net is a **trainable controller**!



# Program execution

Learn to predict the outcome of simple python programs

**Input:**

```
j=8584
for x in range(8):
    j+=920
b=(1500+j)
print((b+7567))
```

**Target:** 25011.**Input:**

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else 5308)
```

**Target:** 12184.

Seems easy? This is what the net sees!

**Input:**

```
vqppkn
sqdvfljmnc
y2vxdddsepnimcbvubkomhrpliibtwztbljipcc
```

**Target:** hkhpg