Machine Learning

Conditional Random Fields (CRF)

Task description – more formal

- Given a sentence and its tokens t_i, assign a single label $l \in L$ with L being the tagset (e.g. Penn Treebank, STTS) to every t_i
- This is a **structural problem**, where our input is a sequence (“a list of tokens”) and the output is a sequence (“a list of labels”), and:
  - Both sequences have the same length
    - (This is not the case for OCR or speech recognition)

WORD	tag
the	**DET**
koala	**N**
put	**V**
the	**DET**
keys	**N**
on	**P**
the	**DET**
table	**N**

From MEMM to CRF

- We saw, that we still have issues with local normalization

- We repeatedly apply the following MaxEnt classifier:

$$p(t|w) = \prod_i \frac{\text{score}(w, t_i, t_{i-1})}{\sum_{t_i, t_{i-1} \in L} \text{score}(w, t_i, t_{i-1})}$$

Since we apply local models, the denominator has just „a few“ terms (in our case quadratically many)

From MEMM to CRF

- If we would simply increase the order of our MEMM, the denominator gets bigger and more problematic to calculate, and we would end up with

- Applying a gigantic MaxEnt model:

$$p(t|w) = \frac{\mathrm{score}(w,t)}{\sum_{t \in seq} \mathrm{score}(w,t)}$$

We have to sum and score every single sequence! And this changes with every new input we are trying to predict

From MEMM to CRF

- The good things first:
  - Finding the best performing sequence is still feasible:

$$\hat{t} = argmax_{t \in seq} \frac{\text{score}(w,t)}{\sum_{t \in seq} \text{score}(w,t)} = argmax_{t \in seq} \text{score}(w,t)$$

- Because we can ignore the denominator entirely!
  - Thanks Viterbi!

From MEMM to CRF

- But we can not ignore this during training!

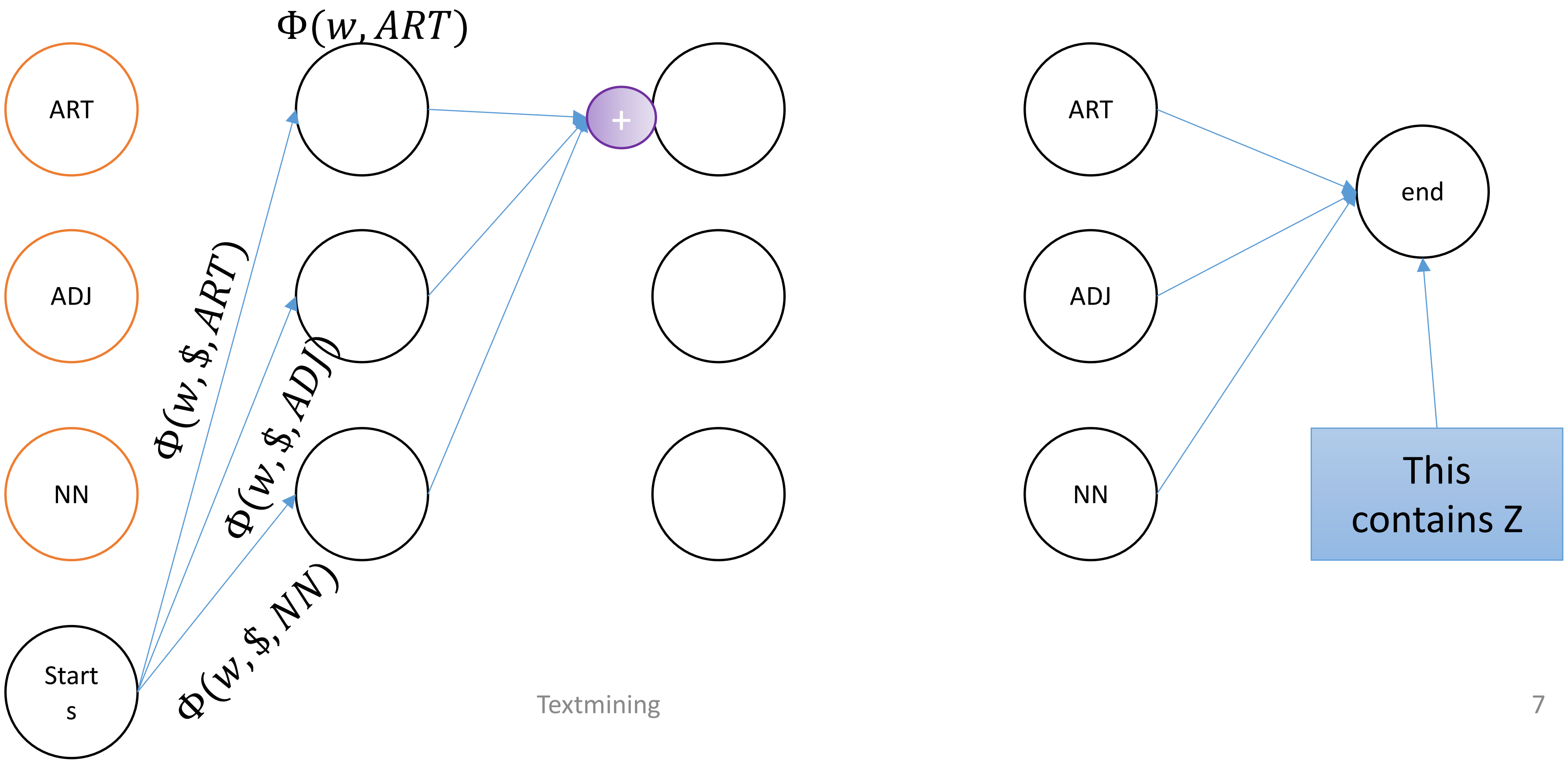
- Let us focus on the denominator: $Z = \sum_{t \in seq} \text{score}(w, t)$

- This reads as: „The sum of the score of all sequences“

  - This can be solved via dynamic programming with the „forward algorithm“

Forward Algorithm

- The forward algorithm is almost the same as the Viterbi, but instead of taking the **max** of all incoming arcs, we **sum**, them!



Conditional Random Field

- A Conditional Random Field is basically a Maximum Entropy classifier which scores a structure (in exactly the same way as a MEMM)
- But: Has a different way of normalizing the score into a probability distribution
  - This comes only into play during the training of a CRF
- CRFs are really powerful classifiers, which can be compared using the integrated templates $\Phi(w, t)$ or in general $\Phi(X, Y)$

Conditional Random Fields (CRF)

- A general procedure (for sequences):
  1. Define a set of feature templates $\Phi(X, Y)$
  2. Create the resulting transducer
  3. Assign the templates to the edges and nodes of the transducer
  4. Unroll the transducer for a given example
  5. Decode using the Viterbi algorithm

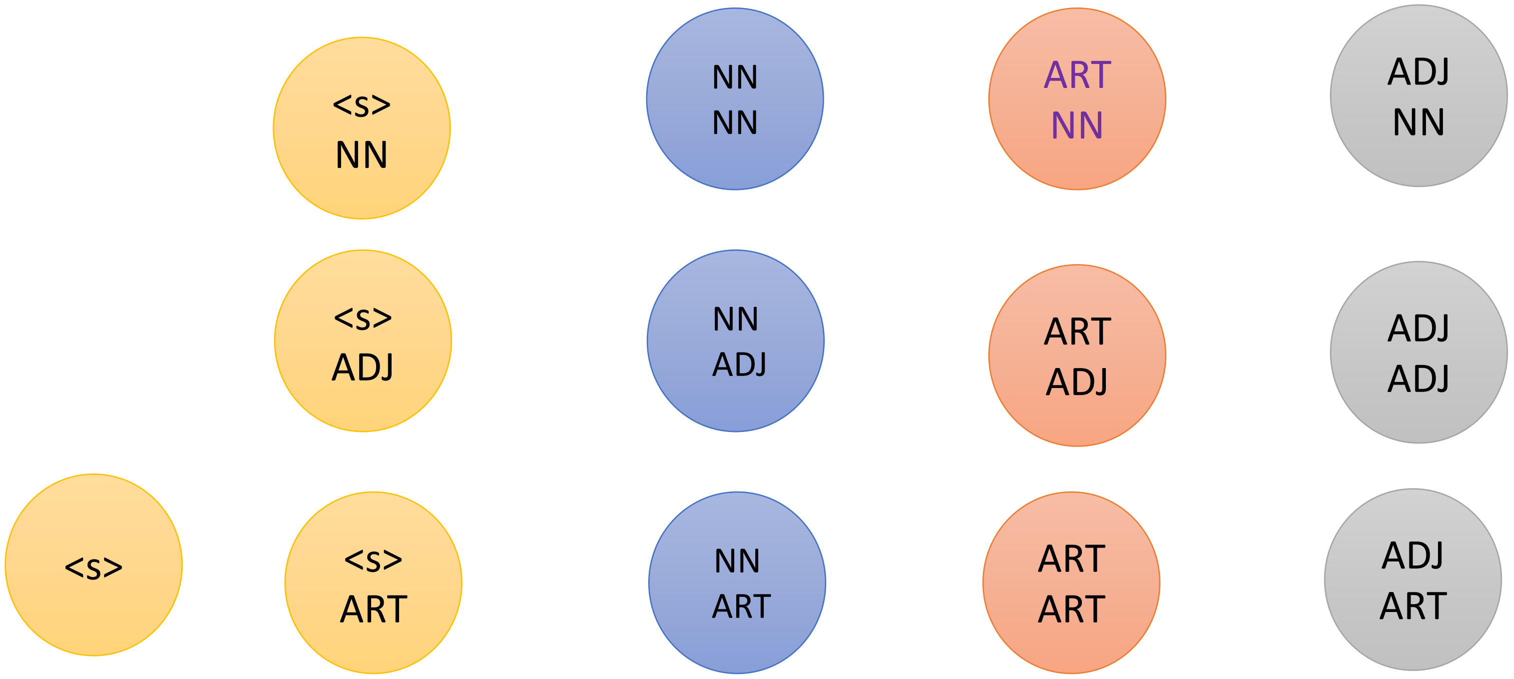
Conditional Random Fields (CRF)

1. Define a set of feature templates $\Phi(X, Y)$, e.g.:
   - $\Phi(x, y_t)$ (Node-template)
   - $\Phi(x, y_t, y_{t-1})$ (Edge-template, **order-1**)
   - $\Phi(x, y_t, y_{t-1}, y_{t-2})$ (Edge-template, **order-2**)

Conditional Random Fields (CRF)

2. Create the resulting transducer

➔ Order 2 template now creates states of tuples

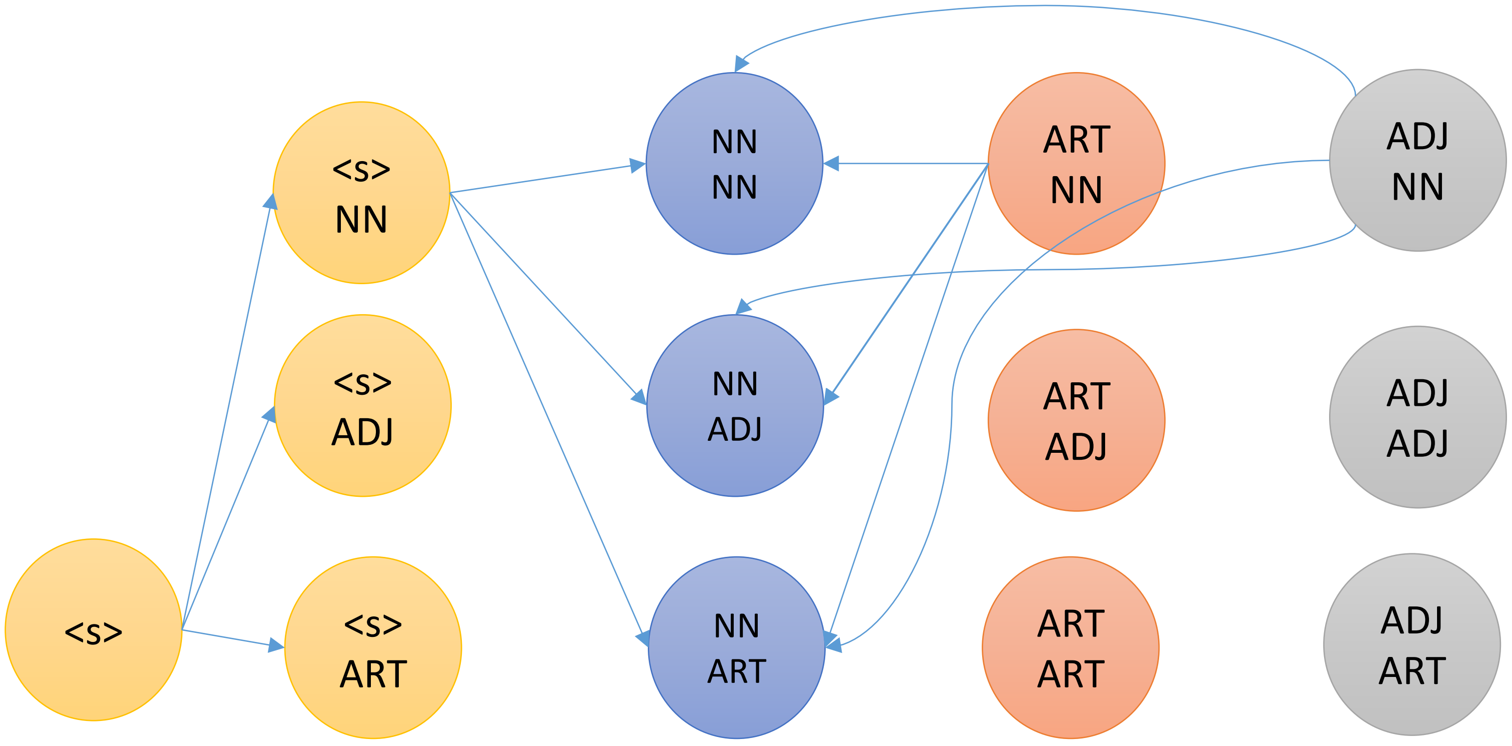


ART NN: We have transitioned from a state with suffix ART into NN

Conditional Random Fields (CRF)

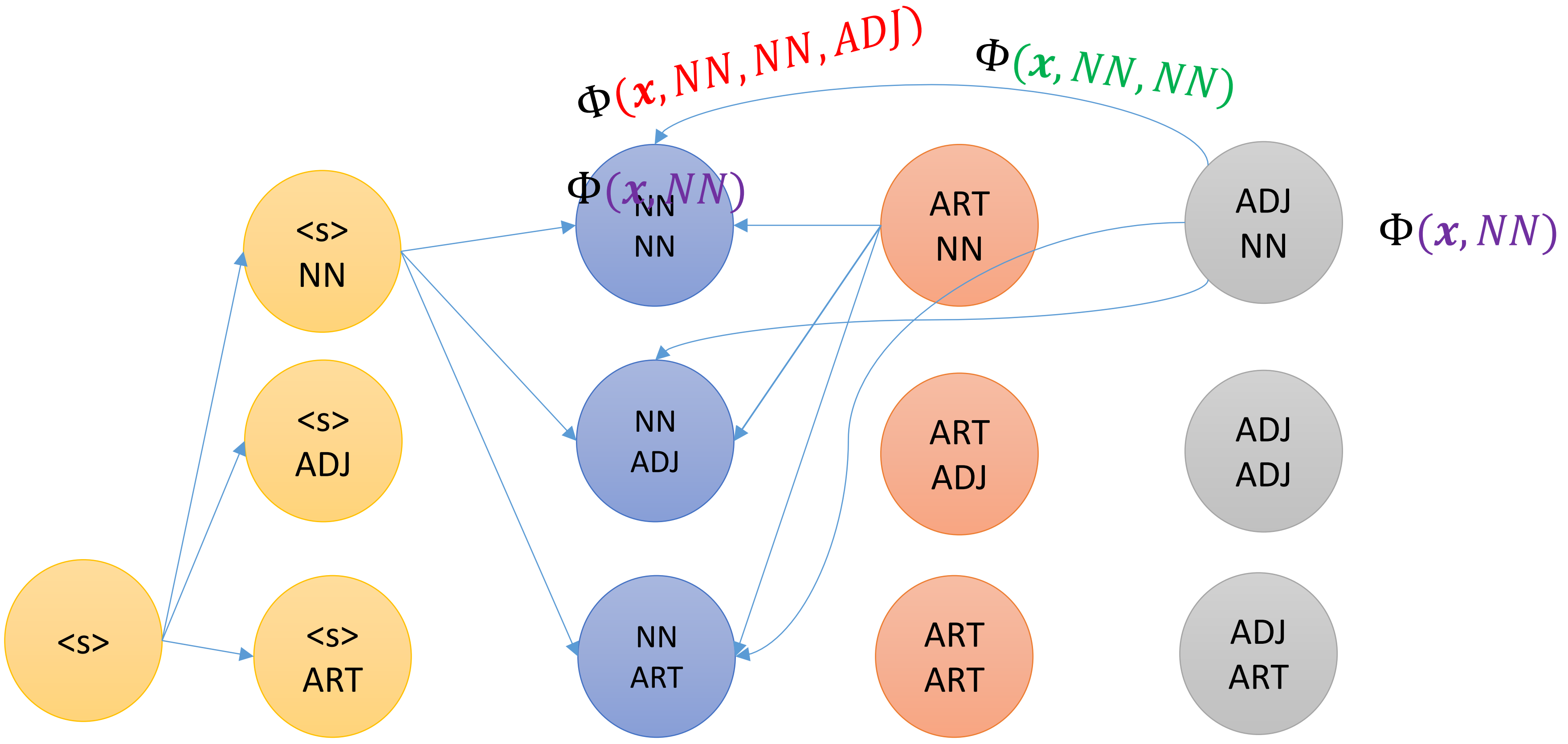
2. Create the resulting transducer

   ➔Not all transitions viable! (only subset drawn!)



Conditional Random Fields (CRF)

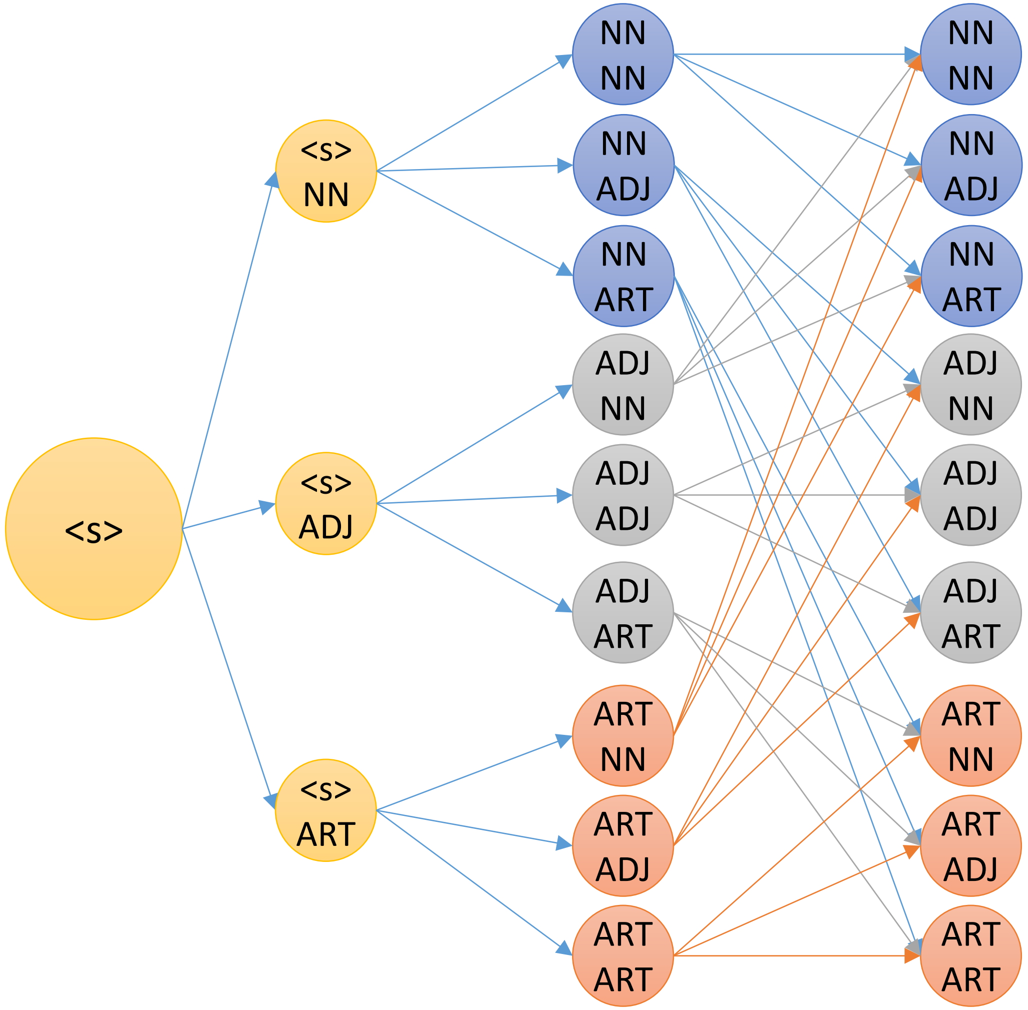
3. Assign the templates to the edges and nodes of the transducer



$\Phi(x, y_t)$
$\Phi(x, y_t, y_{t-1})$
$\Phi(x, y_t, y_{t-1}, y_{t-2})$

Conditional Random Fields (CRF)

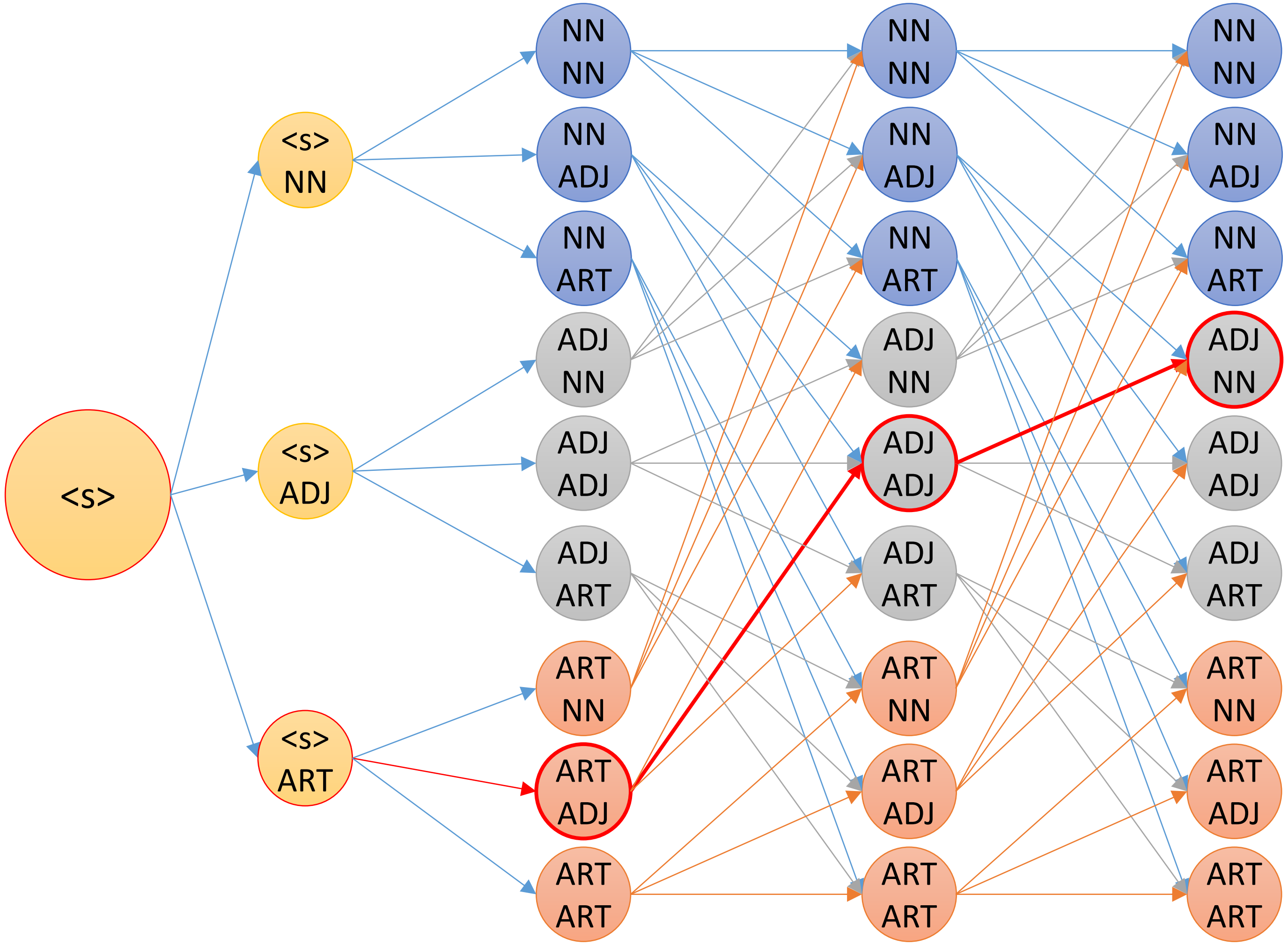
4. Unroll the transducer for a given example



...

Conditional Random Fields (CRF)

5. Decode using the Viterbi algorithm (only solution marked)



Conditional Random Fields (CRF)

- Templates are very powerful and still relevant for Neural Architectures, since they **define** the architecture

- Other useful templates:
  - Multitask-template (learn task t and u at the same time): $\Phi(x, Y_t, Y_u)$
    - „Score if I set label A for task t and label B for task u“

  - Semi-(Markov)-Template: $\Phi(x, y_t, N)$
    - N gives you the amount of steps you have already seen the label y_t in sequence
    - „What is the score to predict ADJ if we have already seen 3 ADJ in a row“

  - Exists-Template: $\Phi(x, y, \vec{b})$
    - With b being a Boolean vector which stores which states we already visited
    - „Have I already seen a verb in my current sequence “

Conditional Random Fields (CRF)

- Parameter Learning:
  - A CRF (as presented here) is a single, but very large MaxEnt model
    - ➔Parameter learning using Gradient Descent
    - ➔Difference is that it applies features in a dynamic fashion

  - I’m not going into detail, how to efficiently calculate the gradient, since this involves numerous numerical tricks

  - In essence it makes use of the Forward-Backward Algorithm, which is very similar to the Viterbi
    - In fact the forward algorithm is equal to Viterbi, but the **max** is replaced with a **sum**