Hidden Markov Models

Sequence labeling

Problem

- Given a sequence of *tokens*, infer the most probable sequence of *labels* for these tokens.

Examples

- part of speech tagging
- named entity recognition

Example: part-of-speech tagging

Tokens are words, and labels are PoS tags.

PRON	VERB	DET	PROPN	NOUN
I	saw	a	Heffalump	today.

PoS tags from Universal Dependencies project

Open class words	
ADJ	adjective
ADV	adverb
INTJ	interjection
NOUN	noun
PROPN	proper noun
VERB	verb

Other	
PUNCT	punctuation
SYM	symbol
X	other

Closed class words	
ADP	adposition
AUX	auxiliary verb
CCONJ	coordinating conjunction
DET	determiner
NUM	numeral
PART	particle
PRON	pronoun
SCONJ	subordinating conjunction

http://universaldependencies.org/

Example: named entity recognition

Once upon a time, a very long time ago now, about **[last Friday]**, **[Winnie-the-Pooh]** lived in a forest all by himself under the name of **[Sanders]**.

Types of named entities

Any real-world object which may have a proper name:

- persons
- organizations
- locations
- …

Also named entities usually include:

- dates and times
- units
- amounts

Approaches to sequence labeling

1. Rule-based models (example: EngCG tagger)
2. Separate label classifiers for each token
3. Sequence models (HMM, MEMM, CRF)
4. Neural networks

PoS tagging with HMMs

Notation:

$\mathbf{x} = x_1, \ldots x_T$ is a sequence of words (input)

$\mathbf{y} = y_1, \ldots y_T$ is a sequence of their tags (labels)

We need to find the most probable sequence of tags given the sentence:

$$\mathbf{y} = \text{argmax}_{\mathbf{y}}\, p(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}}\, p(\mathbf{x}, \mathbf{y})$$

But first, let us define the model…

Hidden Markov Model

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})\, p(\mathbf{y}) = \prod_{t=1}^{T} p(x_t|y_t) p(y_t|y_{t-1})$$

observable hidden

Markov assumption:

$$p(\mathbf{y}) \approx \prod_{t=1}^{T} p(y_t|y_{t-1})$$

Output independence:

$$p(\mathbf{x}|\mathbf{y}) \approx \prod_{t=1}^{T} p(x_t|y_t)$$

Text generation in HMM

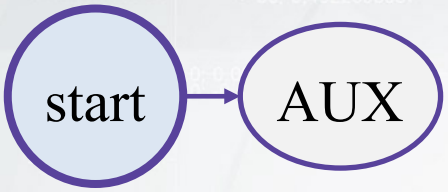
Assume that the text is generated in the following manner:

- One chooses the next PoS tag given the previous tag
- Given the current tag, one generates another word

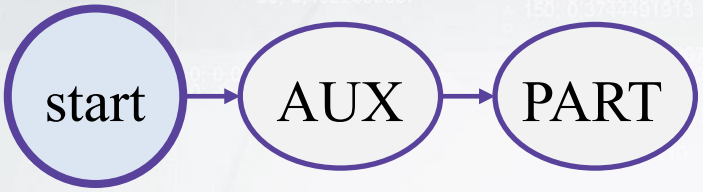
Thus, the neighboring words do not depend on each other, but they depend on the underlying tags.

Text generation in HMM: an example

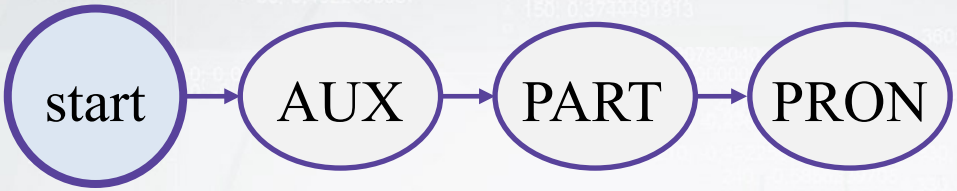
Text generation in HMM: an example



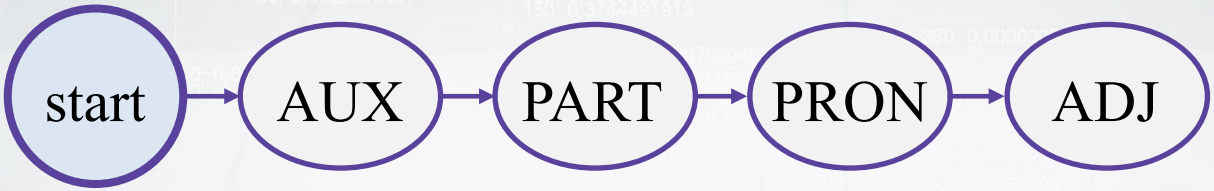
Text generation in HMM: an example



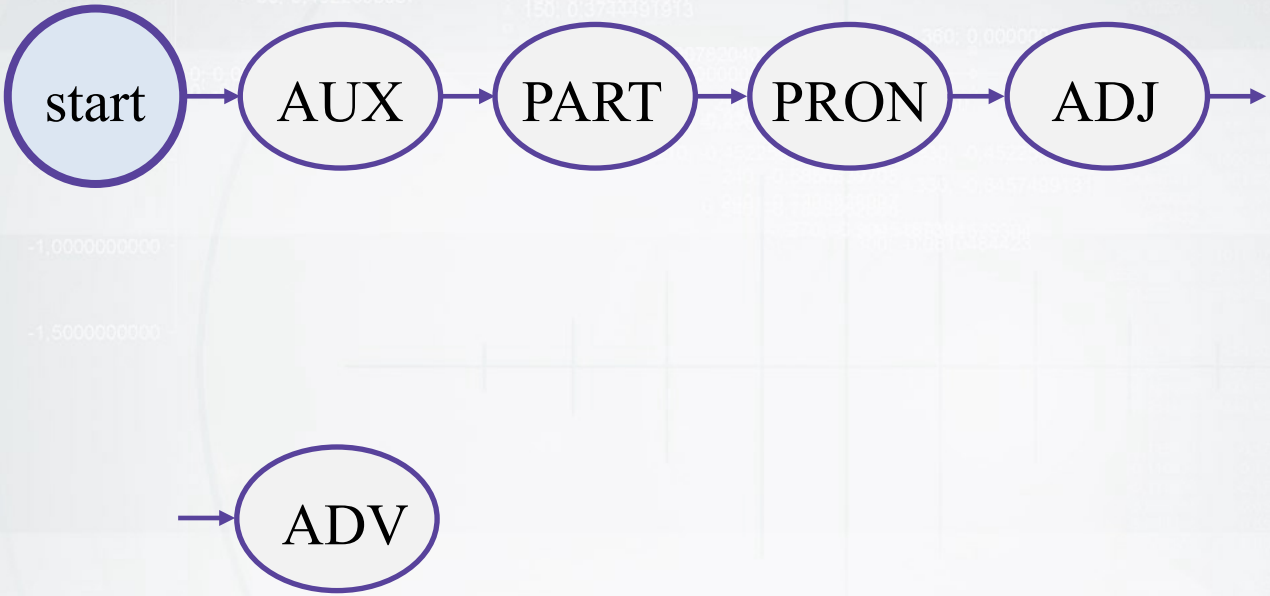
Text generation in HMM: an example



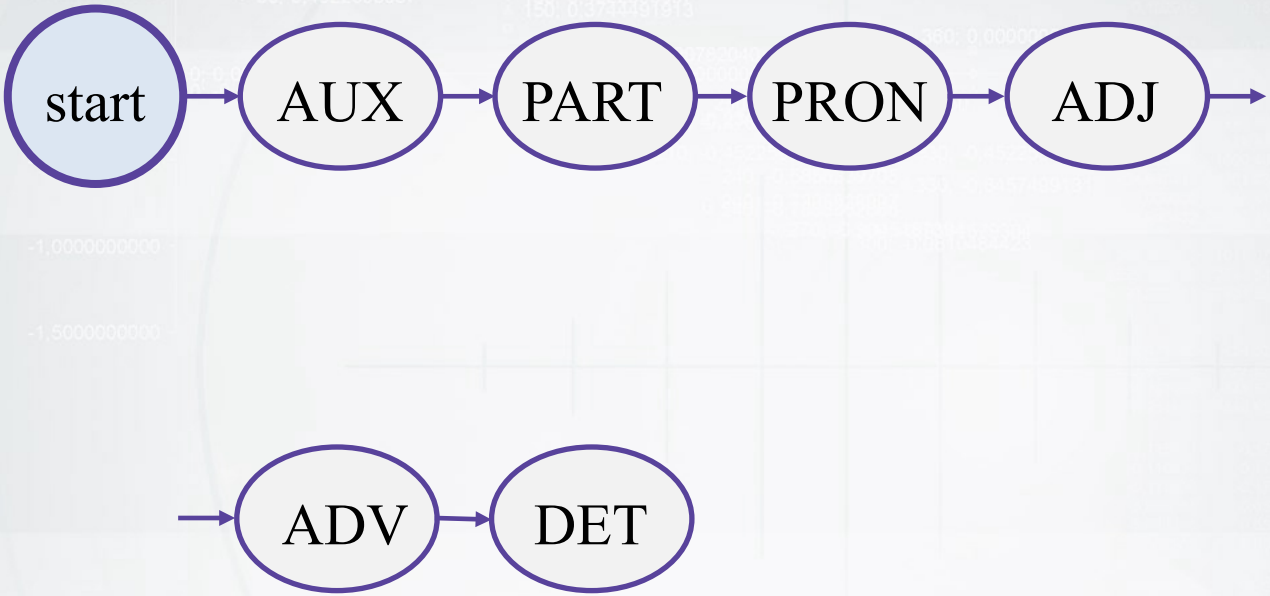
Text generation in HMM: an example



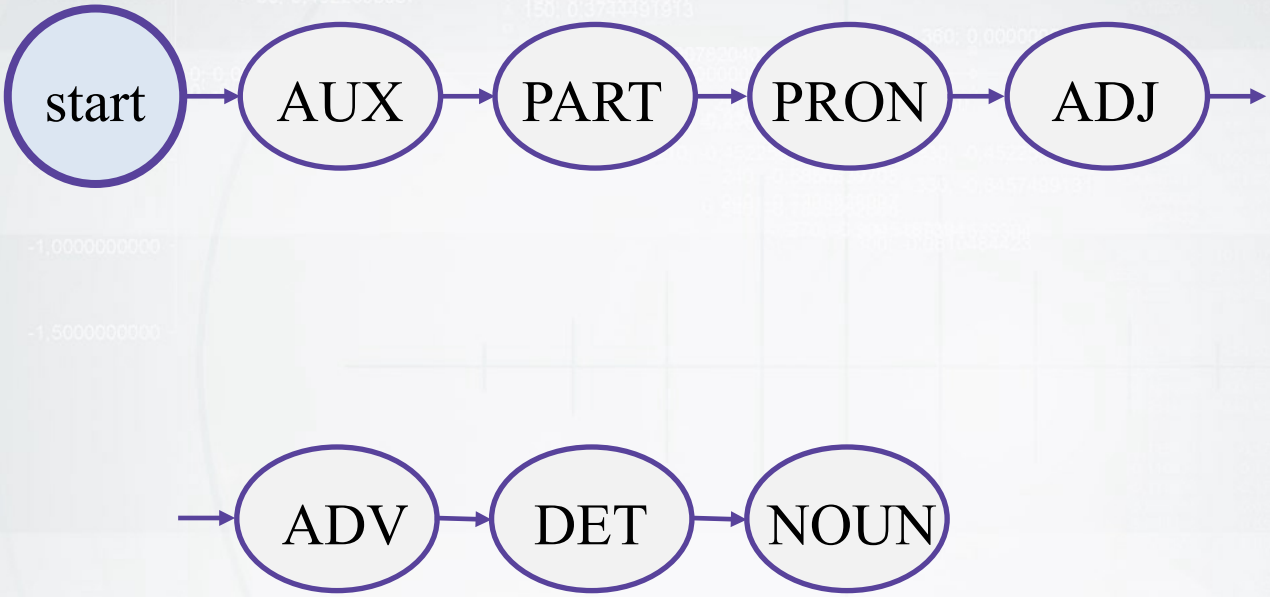
Text generation in HMM: an example



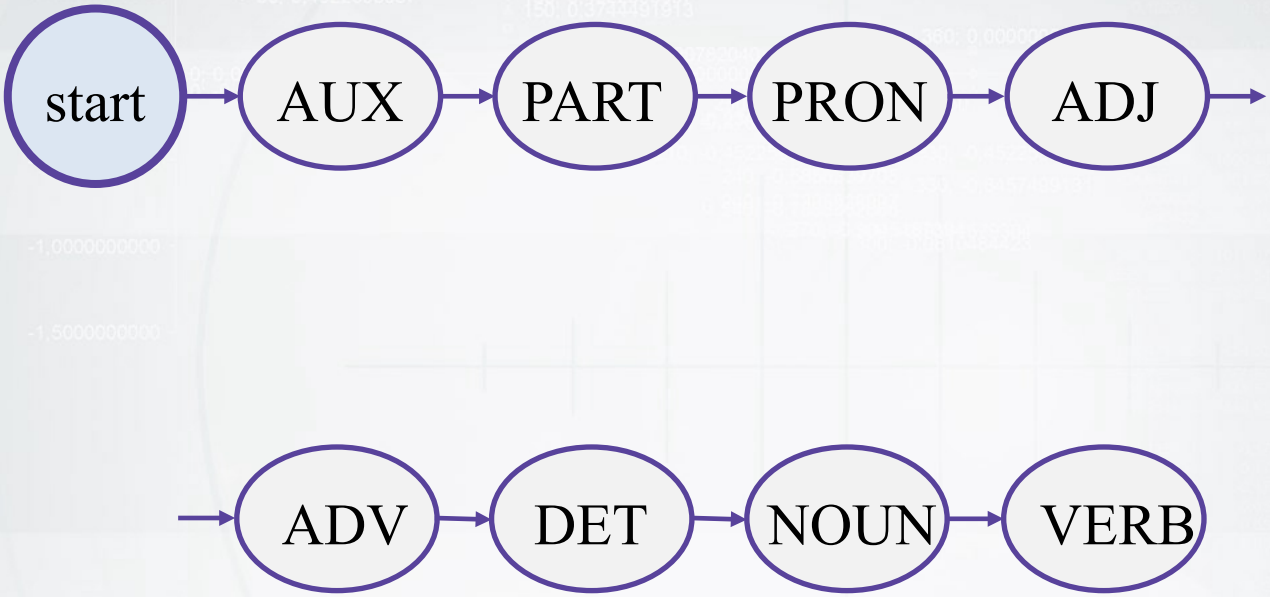
Text generation in HMM: an example



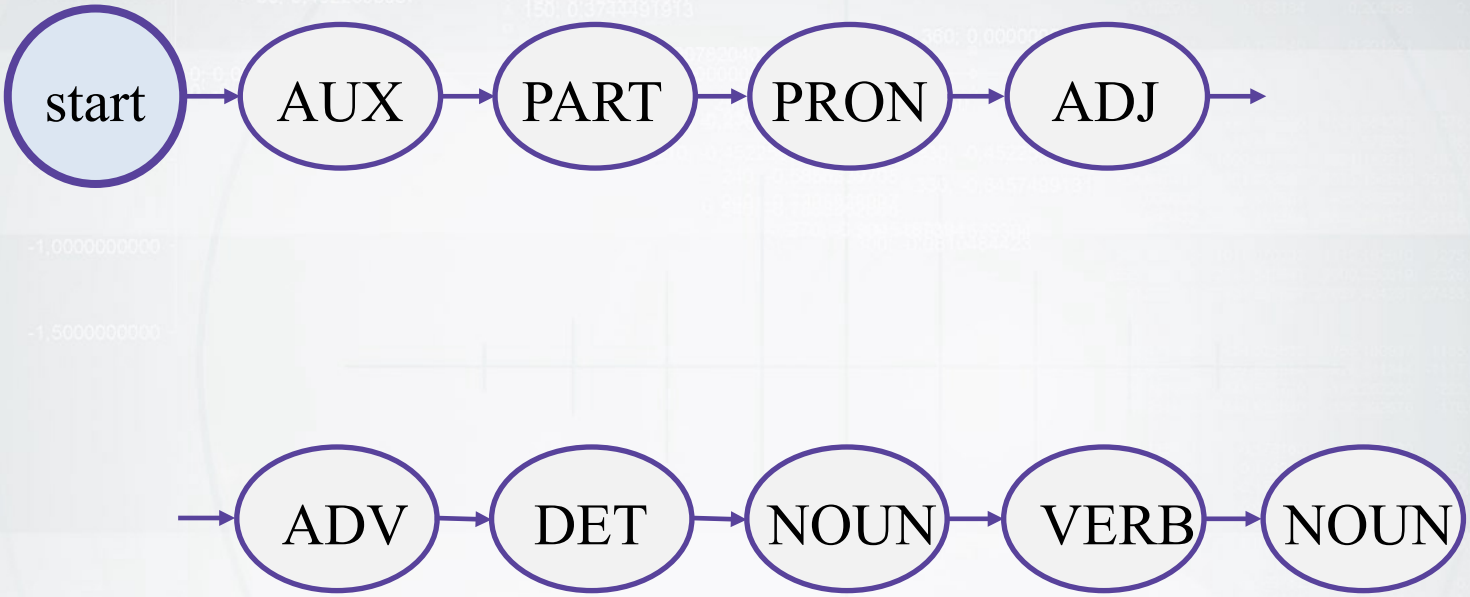
Text generation in HMM: an example



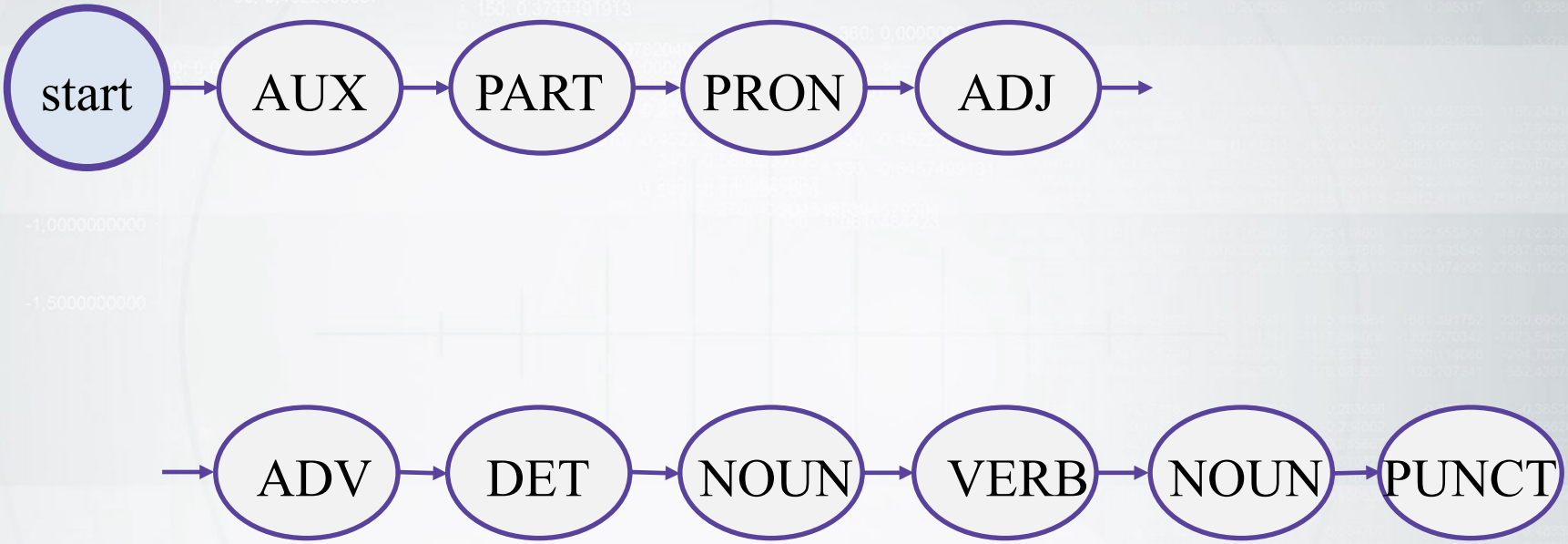
Text generation in HMM: an example



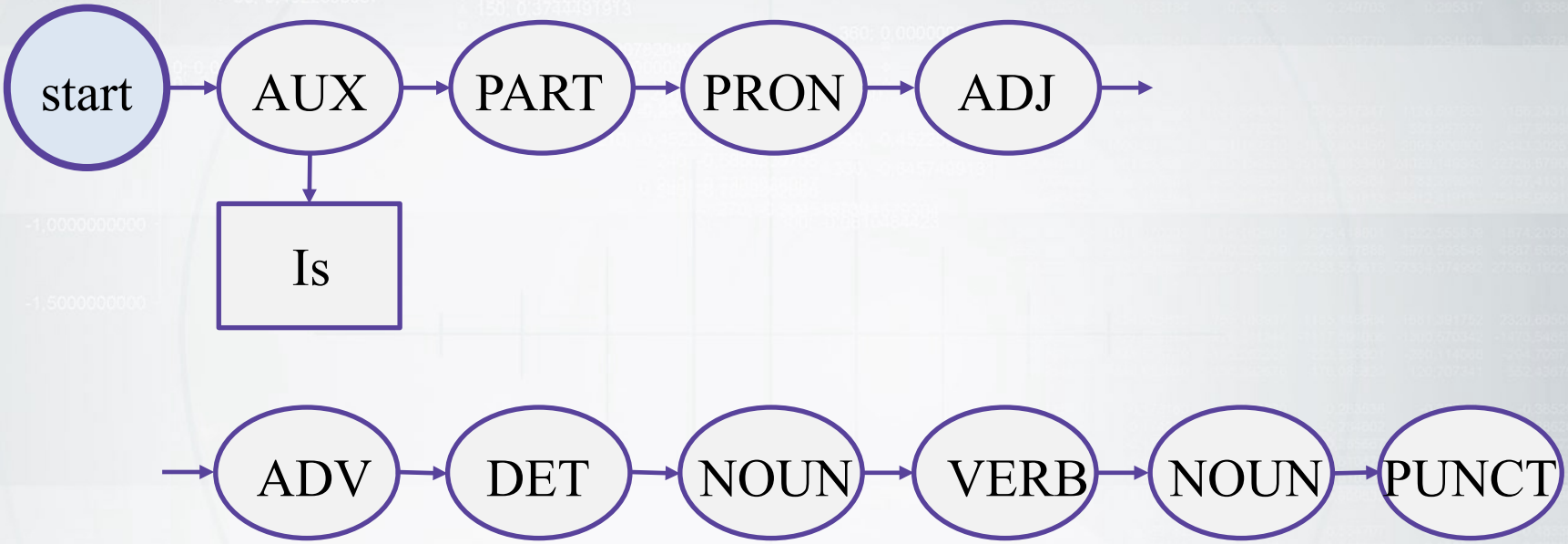
Text generation in HMM: an example



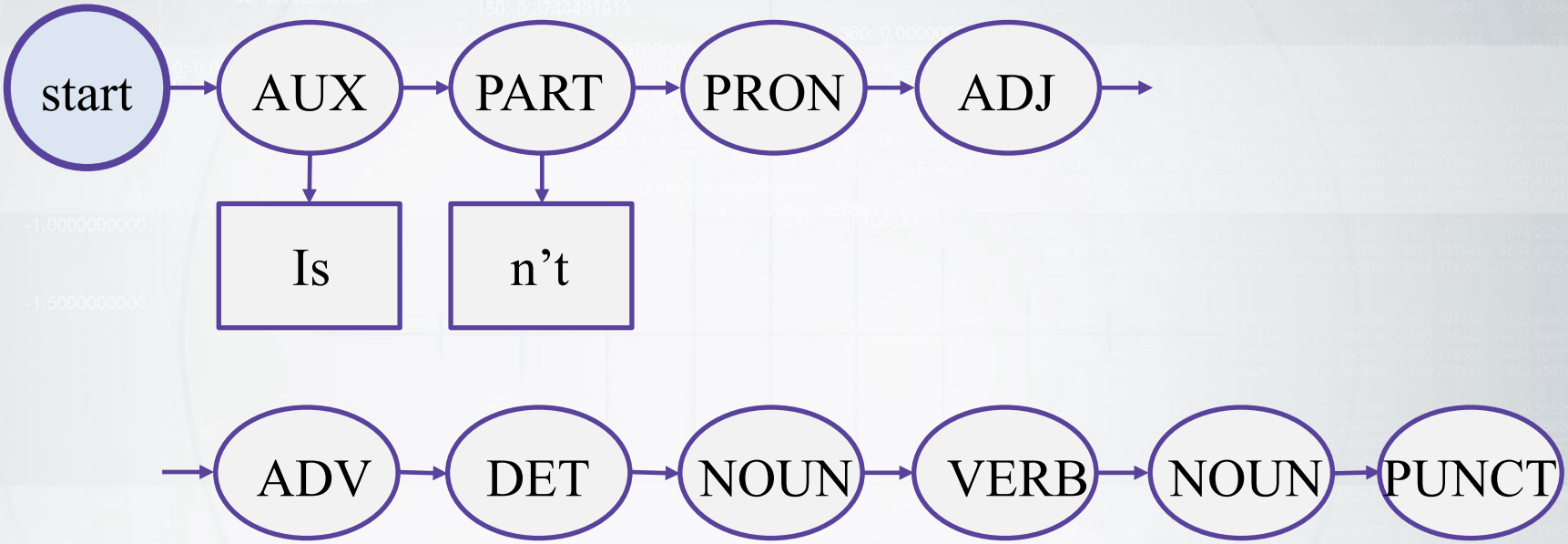
Text generation in HMM: an example



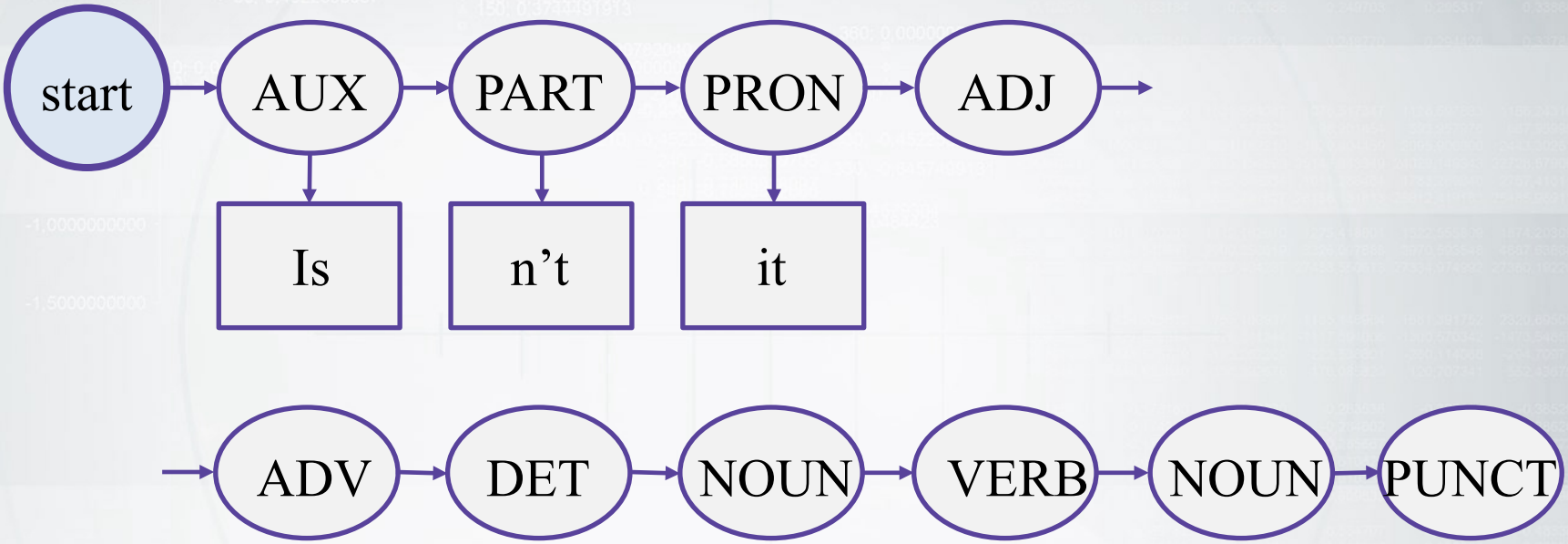
Text generation in HMM: an example



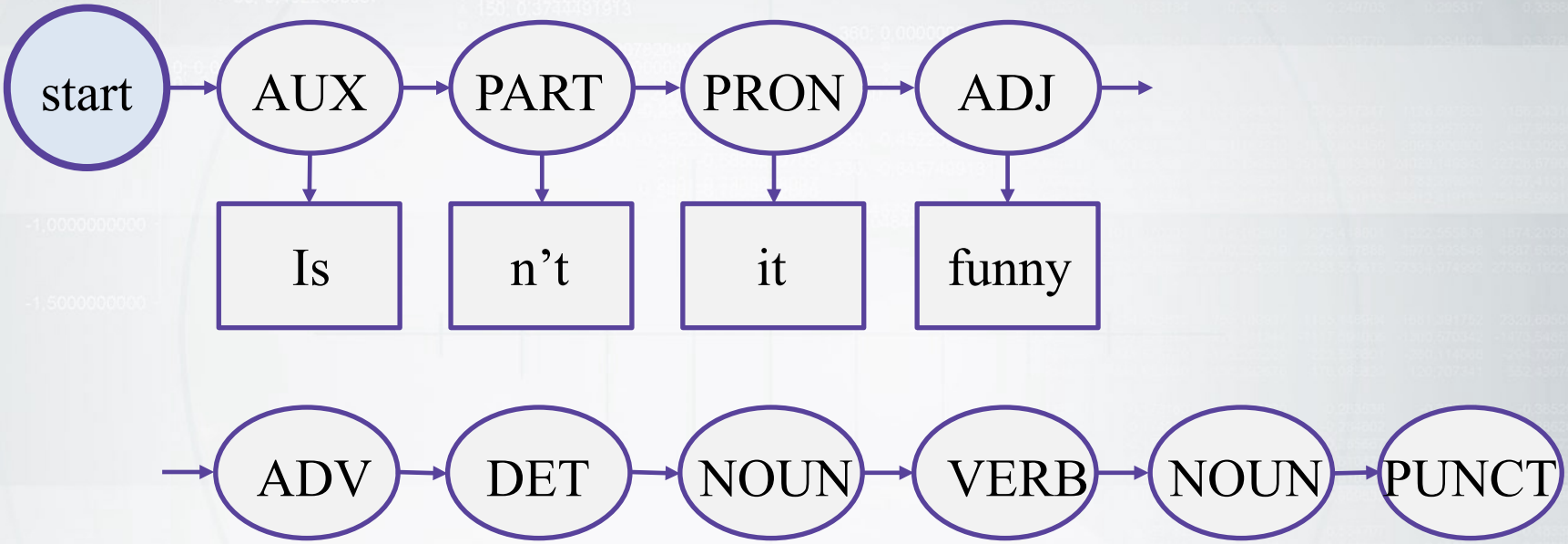
Text generation in HMM: an example



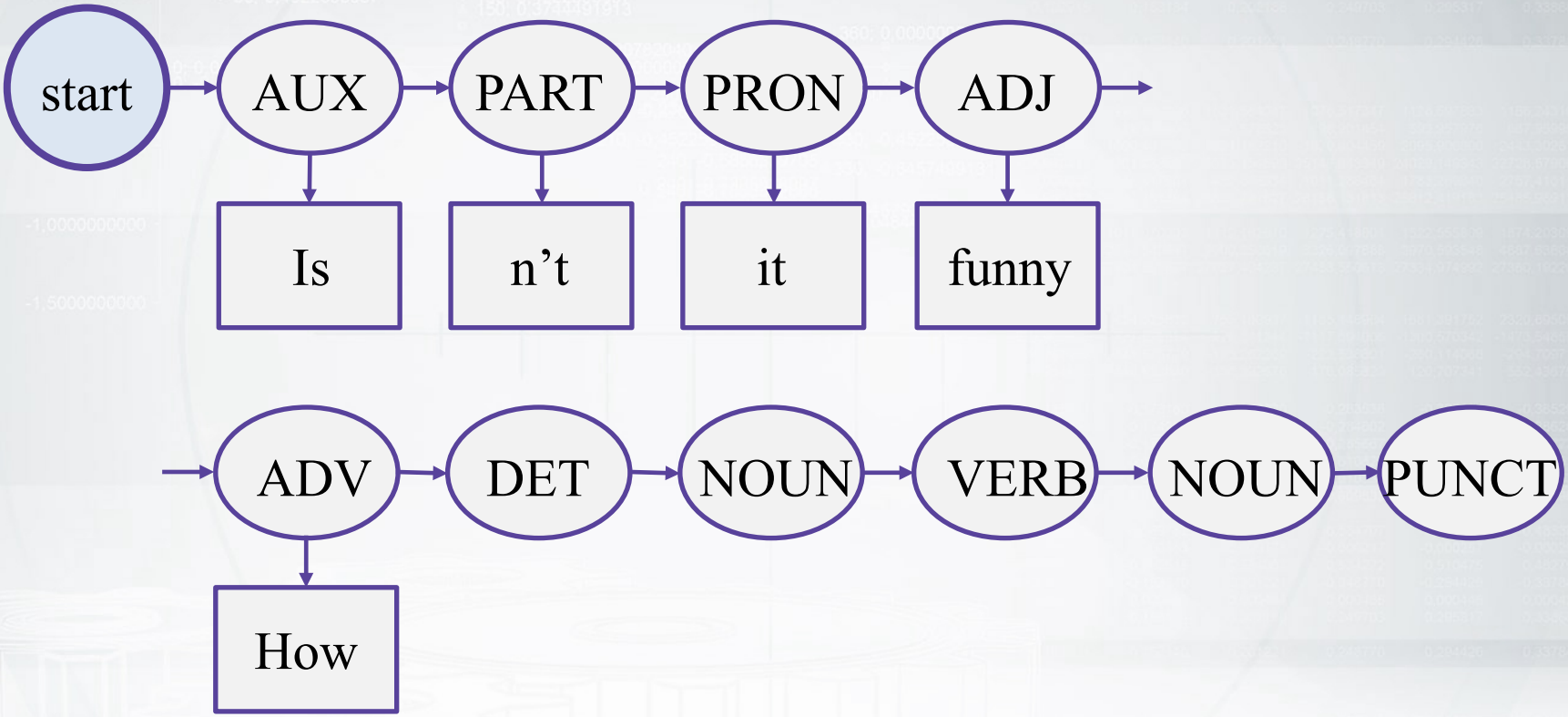
Text generation in HMM: an example



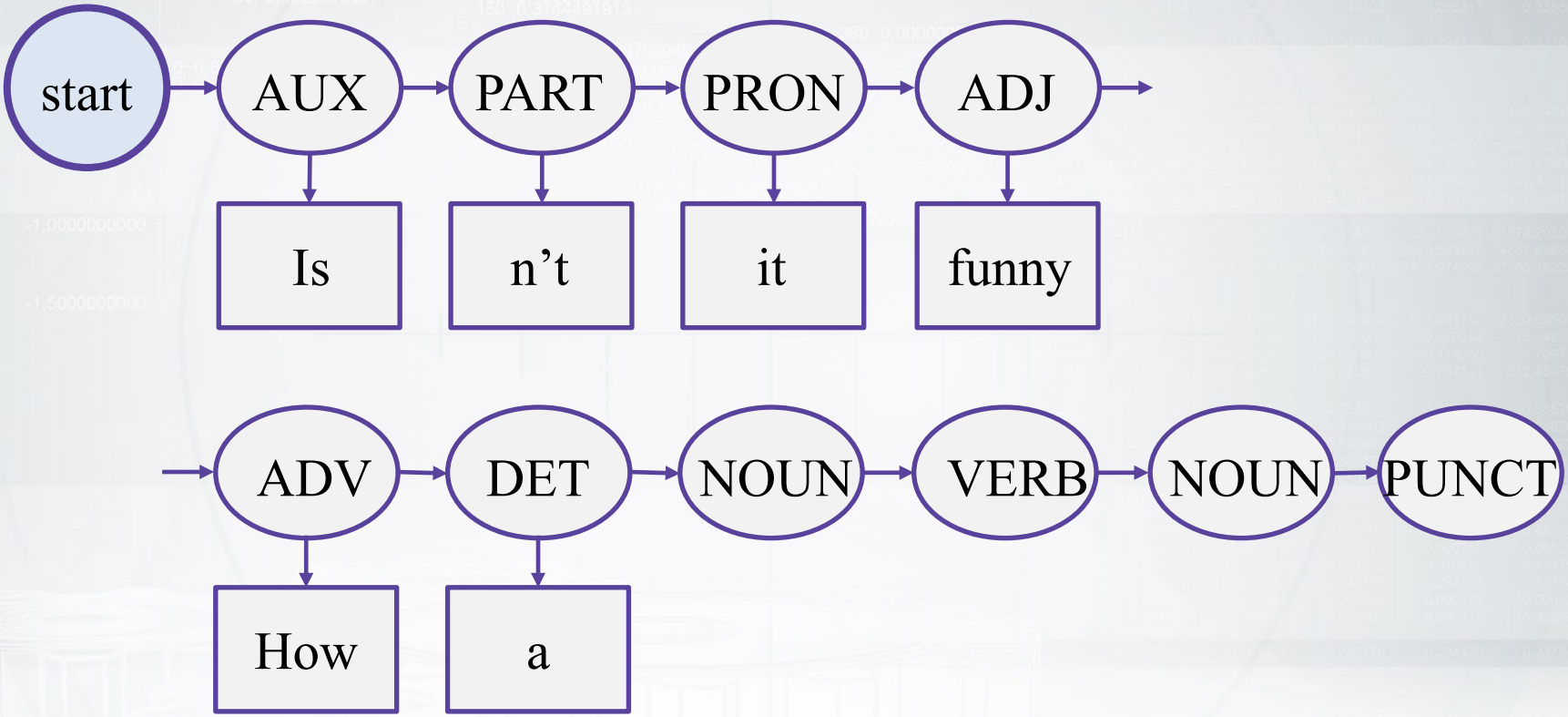
Text generation in HMM: an example



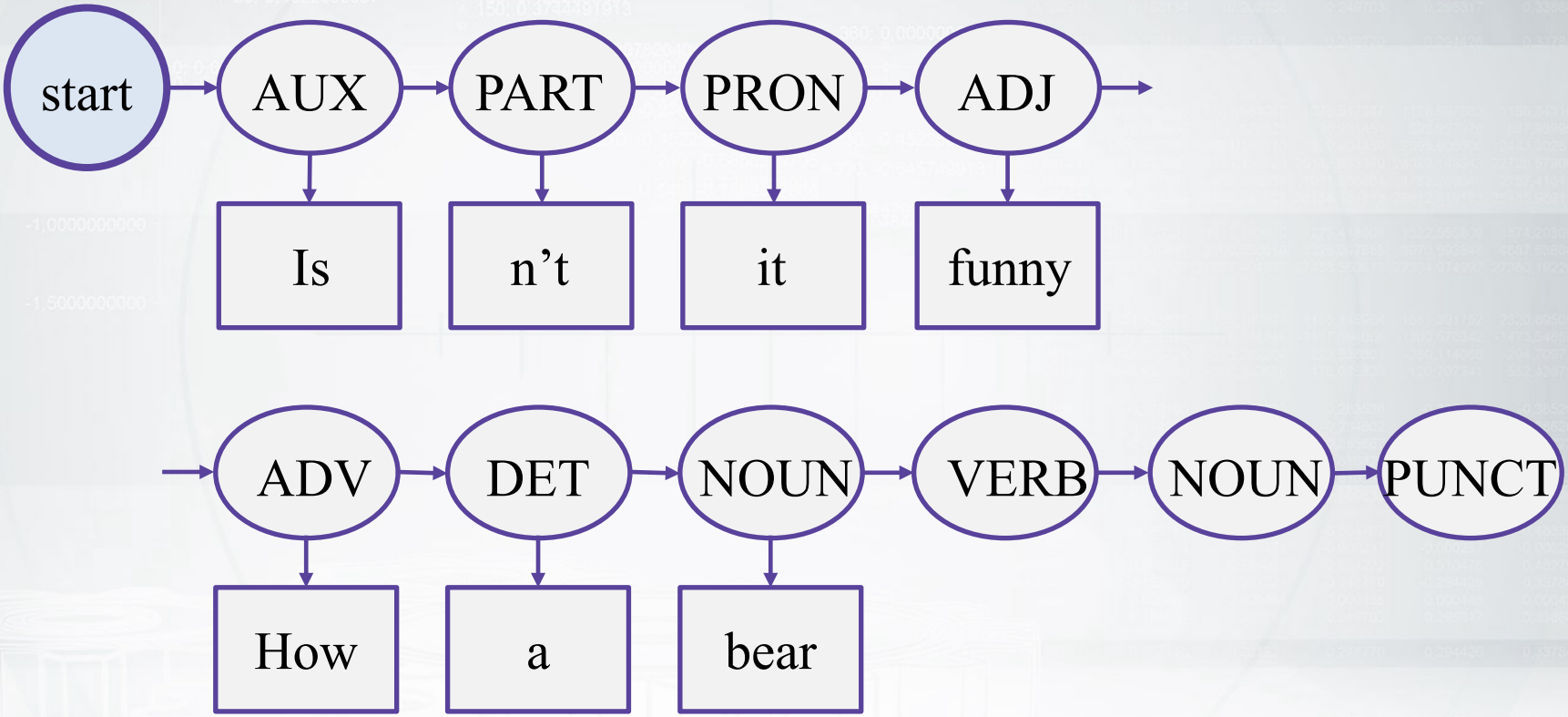
Text generation in HMM: an example



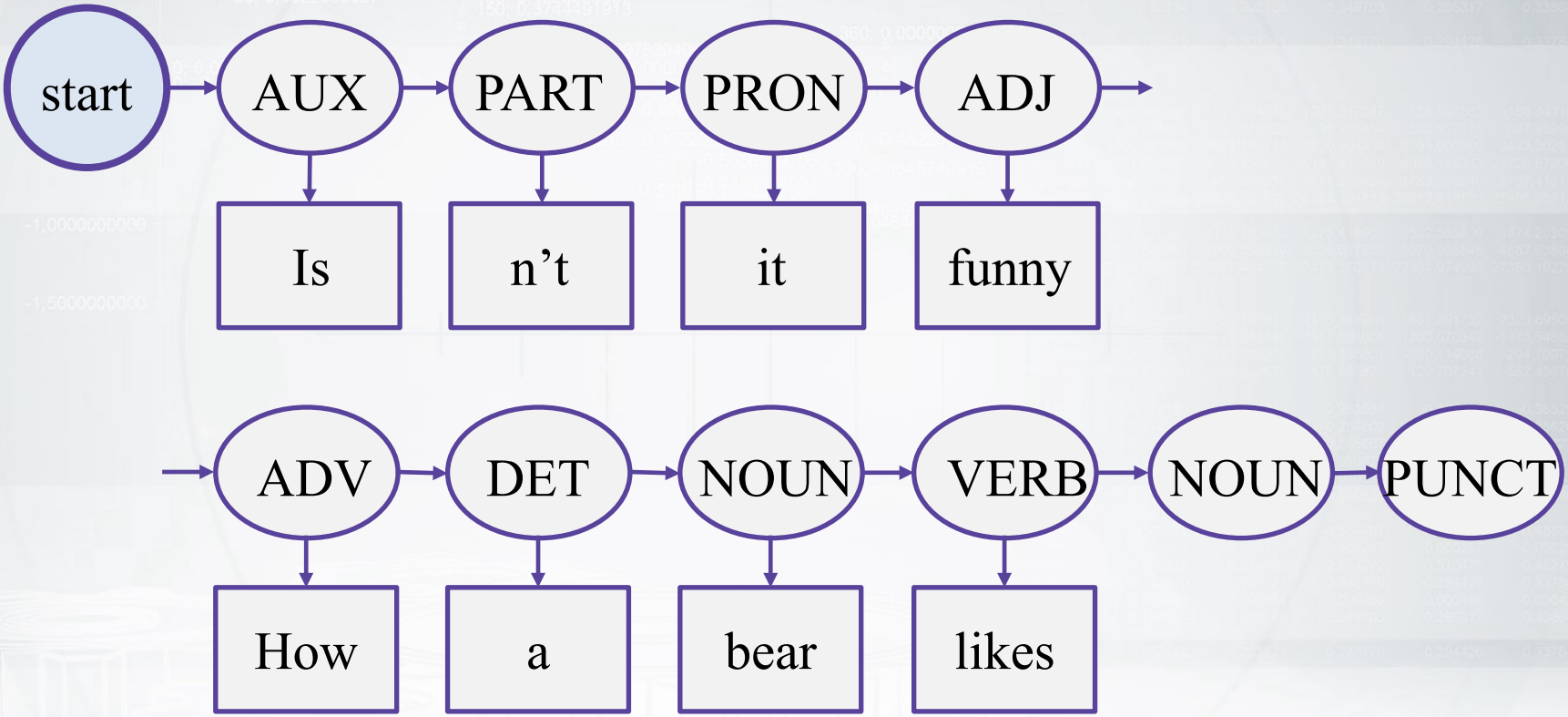
Text generation in HMM: an example



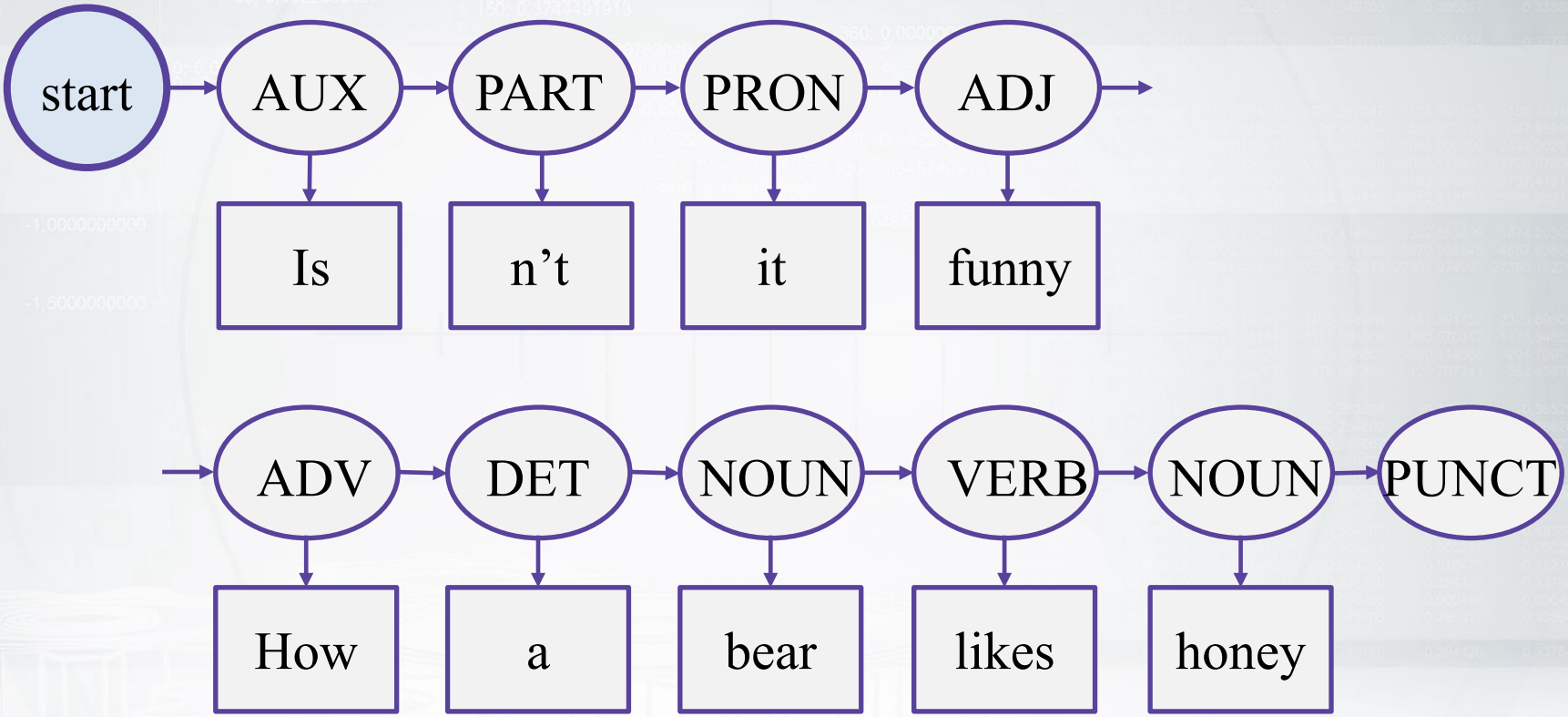
Text generation in HMM: an example



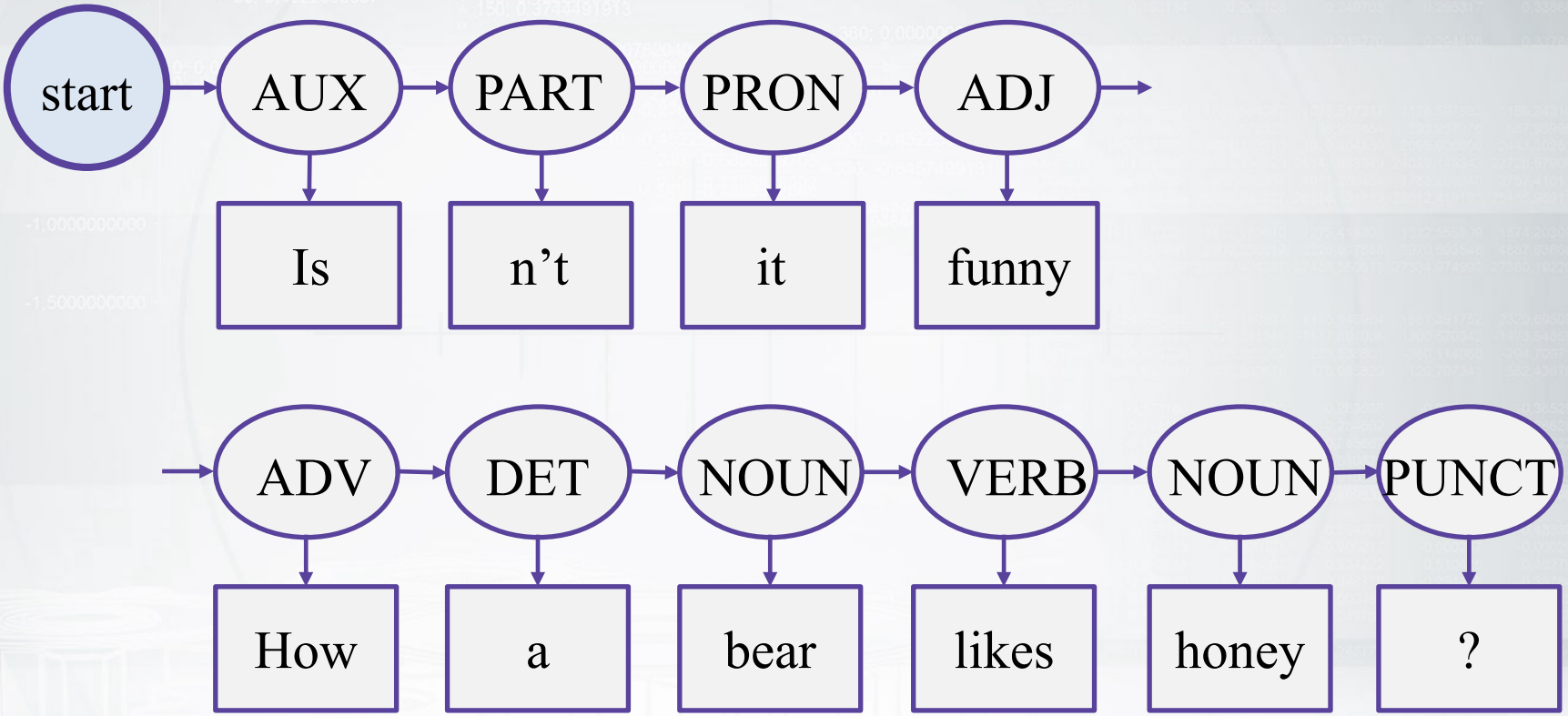
Text generation in HMM: an example



Text generation in HMM: an example



Text generation in HMM: an example



Formal definition of HMM

A Hidden Markov Model is specified by:

1. The set $S = s_1, s_2, \ldots, s_N$ of hidden states
2. The start state s_0
3. The matrix A of transition probabilities: $a_{ij} = p(s_j|s_i)$
4. The set O of possible visible outcomes
5. The matrix B of output probabilities: $b_{kj} = p(o_k|s_j)$

How to train the model?

If we could see the tags in train set, we would count:

$$a_{ij} = p(s_j|s_i) = \frac{c(s_i \rightarrow s_j)}{c(s_i)}$$

$$b_{ik} = p(o_k|s_i) = \frac{c(s_i \rightarrow o_k)}{c(s_i)}$$

Supervised case: MLE

The same in more formal terms (this is MLE!):

$$a_{ij} = p(s_j|s_i) = \frac{\sum_{t=1}^{T} [y_{t-1} = s_i, y_t = s_j]}{\sum_{t=1}^{T} [y_t = s_i]}$$

Note that the corpus is considered as a single sequence of length T with special states between the sentences.

Supervised case: MLE

The same in more formal terms (this is MLE!):

$$a_{ij} = p(s_j|s_i) = \frac{\sum_{t=1}^{T} [y_{t-1} = s_i, y_t = s_j]}{\sum_{t=1}^{T} [y_t = s_i]}$$

Note that the corpus is considered as a single sequence of length T with special states between the sentences.

But we do not see the labels! How to train the model?

Baum-Welch algorithm (a sketch)

E-step: posterior probabilities for hidden variables:

$$p(y_{t-1} = s_i, y_t = s_j)$$

Can be effectively done with dynamic programming (forward-backward algorithm)

M-step: maximum likelihood updates for the parameters:

$$a_{ij} = p(s_j|s_i) = \frac{\sum_{t=1}^{T} p(y_{t-1} = s_i, y_t = s_j)}{\sum_{t=1}^{T} p(y_t = s_i)}$$