Natural Language Processing

SoSe 2017

Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

# Part-of-speech tagging

*Dr. Mariana Neves*

*May 22nd, 2017*

# Part-of-Speech (POS) Tags

- Also known as:
  - Part-of-speech tags, lexical categories, word classes, morphological classes, lexical tags

Plays$_{\text{[VERB]}}$ well$_{\text{[ADVERB]}}$ with$_{\text{[PREPOSITION]}}$ others$_{\text{[NOUN]}}$

Plays$_{\text{[VBZ]}}$ well$_{\text{[RB]}}$ with$_{\text{[IN]}}$ others$_{\text{[NNS]}}$

# Examples of POS tags

- **Noun**: book/books, nature, Germany, Sony
- **Verb**: eat, wrote
- **Auxiliary**: can, should, have
- **Adjective**: new, newer, newest
- **Adverb**: well, urgently
- **Number**: 872, two, first
- **Article/Determiner**: the, some
- **Conjuction**: and, or
- **Pronoun**: he, my
- **Preposition**: to, in
- **Particle**: off, up
- **Interjection**: Ow, Eh

# Motivation: Speech Synthesis

- Word „content“
  - „Eggs have a high protein content.“
  - „She was content to step down after four years as chief executive.“

(http://www.thefreedictionary.com/content)

# Motivation: Machine Translation

- e.g., translation from English to German:
  - „I like ...“
    - „Ich mag ....“ (verb)
    - „Ich wie ...“ (preposition)

# Motivation: Syntactic parsing

**Your query**

*I saw the man on the roof*

**Tagging**

```
I/PRP  saw/VBD  the/DT  man/NN  on/IN  the/DT  roof/NN
```

**Parse**

```
(ROOT
  (S
    (NP (PRP I))
    (VP (VBD saw)
      (NP (DT the) (NN man))
      (PP (IN on)
        (NP (DT the) (NN roof))))))
```

(http://nlp.stanford.edu:8080/parser/index.jsp)

# Motivation: Information extraction

- Named-entity recognition (usually nouns)

```
> echo "Inhibition of NF-kappaB activation reversed the anti-apoptotic effect of isochamaejasmin." | ./geniatagger

Inhibition      Inhibition      NN      B-NP    O
of              of              IN      B-PP    O
NF-kappaB       NF-kappaB       NN      B-NP    B-protein
activation      activation      NN      I-NP    O
reversed        reverse         VBD     B-VP    O
the             the             DT      B-NP    O
anti-apoptotic  anti-apoptotic  JJ      I-NP    O
effect          effect          NN      I-NP    O
of              of              IN      B-PP    O
isochamaejasmin isochamaejasmin NN      B-NP    O
.               .               .       O       O
```

(http://www.nactem.ac.uk/tsujii/GENIA/tagger/)

# Motivation: Information extraction

- Relation extraction (triggers are usually verbs)

```
> echo "Inhibition of NF-kappaB activation reversed the anti-apoptotic effect of isochamaejasmin." | ./geniatagger

Inhibition      Inhibition      NN    B-NP    O
of              of              IN    B-PP    O
NF-kappaB       NF-kappaB       NN    B-NP    B-protein
activation      activation      NN    I-NP    O
reversed        reverse         VBD   B-VP    O
the             the             DT    B-NP    O
anti-apoptotic  anti-apoptotic  JJ    I-NP    O
effect          effect          NN    I-NP    O
of              of              IN    B-PP    O
isochamaejasmin isochamaejasmin NN    B-NP    O
.               .               .     O       O
```



(http://www.nactem.ac.uk/tsujii/GENIA/tagger/)

# Open vs. Closed Classes

- Closed
  - limited number of words, do not grow usually
  - e.g., Auxiliary, Article, Determiner, Conjuction, Pronoun, Preposition, Particle, Interjection
- Open
  - unlimited number of words
  - e.g., Noun, Verb, Adverb, Adjective

# POS Tagsets

- There are many parts of speech tagsets
- Tag types
  - Coarse-grained
    - Noun, verb, adjective, ...
  - Fine-grained
    - noun-proper-singular, noun-proper-plural, noun-common-mass, ..
    - verb-past, verb-present-3rd, verb-base, ...
    - adjective-simple, adjective-comparative, ...

# POS Tagsets

- Brown tagset (87 tags)
  - Brown corpus

- C5 tagset (61 tags)
- C7 tagset (146 tags!)

- Penn TreeBank (45 tags) – most used
  - A large annotated corpus of English tagset

# POS Tagging

- The process of assigning a part of speech to each word in a text

- Challenge: words often have more than one POS
  - On my back$_{[NN]}$ (noun)
  - The back$_{[JJ]}$ door (adjective)
  - Win the voters back$_{[RB]}$ (adverb)
  - Promised to back$_{[VB]}$ the bill (verb)

# Ambiguity in POS tags

- 45-tags Brown corpus (word types)
  - Unambiguous (1 tag): 38,857
  - Ambiguous: 8,844
    - 2 tags: 6,731
    - 3 tags: 1,621
    - 4 tags: 357
    - 5 tags: 90
    - 6 tags: 32
    - 7 tags: 6 (well, set, round, open, fit, down)
    - 8 tags: 4 ('s, half, back, a)
    - 9 tags: 3 (that, more, in)

# Baseline method

1. Tagging unambiguous words with the correct label
2. Tagging ambiguous words with their most frequent label
3. Tagging unknown words as a noun

- This method performs around 90% precision

# POS Tagging

- The process of assigning a POS tag to each word in a text. Choosing the best candidate tag for each word.

  - Plays (NNS/VBZ)
  - well (UH/JJ/NN/RB)
  - with (IN)
  - others (NNS)

  - Plays$_{[VBZ]}$ well$_{[RB]}$ with$_{[IN]}$ others$_{[NNS]}$

# Rule-Based Tagging

- Standard approach (two steps):
  1. Dictionaries to assign a list of potential tags
     - Plays (NNS/VBZ)
     - well (UH/JJ/NN/RB)
     - with (IN)
     - others (NNS)
  2. Hand-written rules to restrict to a POS tag
     - Plays (VBZ)
     - well (RB)
     - with (IN)
     - others (NNS)

# Rule-Based Tagging

- Some approaches rely on morphological parsing
  - e.g., EngCG Tagger below

```
REPLACE (<CMH> N NOM SG)
        TARGET (INF)
        IF      (-1C DET/GEN/PP OR CORE-TITLE)
                (NOT -1 (<Rel>) OR (INDEP))
                (NOT 0 ("let") OR OPEN-NOMINAL OR AUXW OR (PREP) OR (CC))
                (NOT 1 (ART) OR (ACC) OR (PRON GEN)) ;
```

The rule replaces all readings containing the INF tag with the tag sequence *<CMH> N NOM SG* if all four context-conditions are satisfied:

....

(http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.972&rep=rep1&type=pdf)

# Sequential modeling

- Many of the NLP techniques should deal with data represented as sequence of items
  - Characters, Words, Phrases, Lines, …

- e.g., for part-of-speech tagging
  - I$_{\text{[PRP]}}$ saw$_{\text{[VBP]}}$ the$_{\text{[DT]}}$ man$_{\text{[NN]}}$ on$_{\text{[IN]}}$ the$_{\text{[DT]}}$ roof$_{\text{[NN]}}$ .

- e.g., for named-entity recognition
  - Steven$_{\text{[PER]}}$ Paul$_{\text{[PER]}}$ Jobs$_{\text{[PER]}}$ ,$_{\text{[O]}}$ co-founder$_{\text{[O]}}$ of$_{\text{[O]}}$ Apple$_{\text{[ORG]}}$ Inc$_{\text{[ORG]}}$ ,$_{\text{[O]}}$ was$_{\text{[O]}}$ born$_{\text{[O]}}$ in$_{\text{[O]}}$ California$_{\text{[LOC]}}$.

# Sequential modeling

- Making a decision based on:
  - Current Observation:
    - Word ($W_0$): „35-years-old“
    - Prefix, Suffix: „computation“ → „comp“, „ation“
    - Lowercased word: „New“ → „new“
    - Word shape: „35-years-old“ → „d-a-a“
  - Surrounding observations
    - Words ($W_{+1}$, $W_{-1}$)
  - Previous decisions
    - POS tags ($T_{-1}$, $T_{-2}$)

# Sequential modeling

- Greedy inference
  - Start in the beginning of the sequence
  - Assign a label to each item using the classifier
  - Using previous decisions as well as the observed data

# Sequential modeling

- Beam inference
  - Keeping the top k labels in each position
  - Extending each sequence in each local way
  - Finding the best k labels for the next position

# Hidden Markov Model (HMM)

- Finding the best sequence of tags ($t_1 \ldots t_n$) that corresponds to the sequence of observations ($w_1 \ldots w_n$)

- Probabilistic View
  - Considering all possible sequences of tags
  - Choosing the tag sequence from this universe of sequences, which is most probable given the observation sequence

$$\hat{t}_1^n = argmax_{t_1^n} P\left(t_1^n | w_1^n\right)$$

# Using the Bayes Rule

$$\hat{t}_1^n = argmax_{t_1^n} P\left(t_1^n \middle| w_1^n\right)$$

$$P\left(A \middle| B\right) = \frac{P\left(B \middle| A\right) \cdot P\left(A\right)}{P\left(B\right)}$$

$$P\left(t_1^n \middle| w_1^n\right) = \frac{P\left(w_1^n \middle| t_1^n\right) \cdot P\left(t_1^n\right)}{P\left(w_1^n\right)}$$

$$\hat{t}_1^n = argmax_{t_1^n} \underbrace{P\left(w_1^n \middle| t_1^n\right)}_{\text{likelihood}} \cdot \underbrace{P\left(t_1^n\right)}_{\text{prior probability}}$$

# Using Markov Assumption

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(w_1^n | t_1^n) \cdot P(t_1^n)$$

$$P(w_1^n | t_1^n) \simeq \prod_{i=1}^{n} P(w_i | t_i)$$ (it depends only on its POS tag and independent of other words)

$$P(t_1^n) \simeq \prod_{i=1}^{n} P(t_i | t_{i-1})$$ (it depends only on the previous POS tag, thus, bigram)

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) \cdot P(t_i | t_{i-1})$$

# Two Probabilities

- The tag transition probabilities: $P(t_i|t_{i-1})$
  - Finding the likelihood of a tag to proceed by another tag
  - Similar to the normal bigram model

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

# Two Probabilities

- The word likelihood probabilities: $P(w_i|t_i)$
  - Finding the likelihood of a word to appear given a tag

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

# Two Probabilities

I$_{\text{[PRP]}}$ saw$_{\text{[VBP]}}$ the$_{\text{[DT]}}$ man$_{\text{[NN?]}}$ on$_{\text{[]}}$ the$_{\text{[]}}$ roof$_{\text{[]}}$ .

$$P\left([NN] \| [DT]\right) = \frac{C\left([DT],[NN]\right)}{C\left([DT]\right)}$$

$$P\left(man | [NN]\right) = \frac{C\left([NN], man\right)}{C\left([NN]\right)}$$

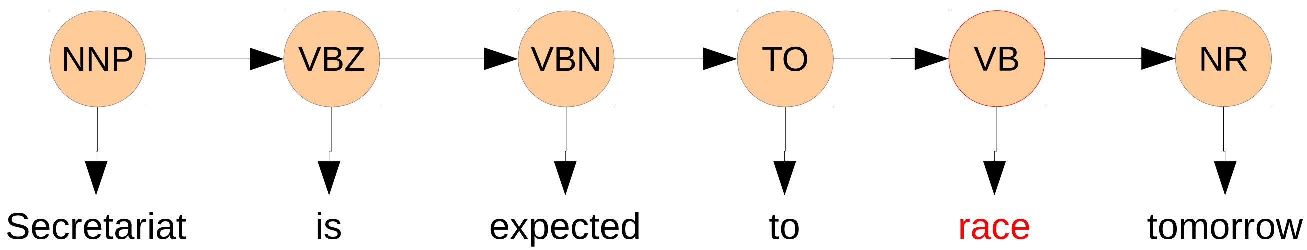
# Ambiguity in POS tagging

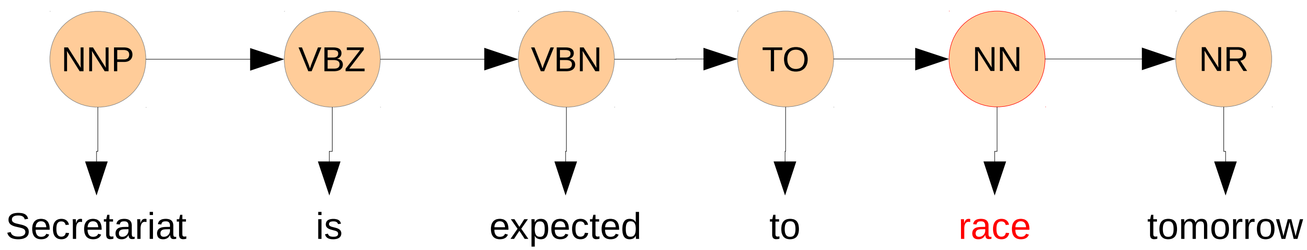
Secretariat$_{[NNP]}$ is$_{[VBZ]}$ expected$_{[VBN]}$ to$_{[TO]}$ race$_{[VB]}$ tomorrow$_{[NR]}$ .

People$_{[NNS]}$ inquire$_{[VB]}$ the$_{[DT]}$ reason$_{[NN]}$ for$_{[IN]}$ the$_{[DT]}$ race$_{[NN]}$ .

# Ambiguity

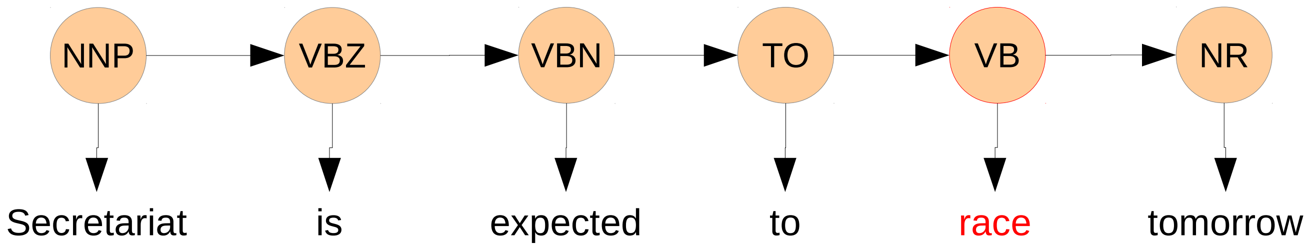
Secretariat$_{[NNP]}$ is$_{[VBZ]}$ expected$_{[VBN]}$ to$_{[TO]}$ race$_{[?]}$ tomorrow$_{[NR]}$ .



# Ambiguity

Secretariat$_{[NNP]}$ is$_{[VBZ]}$ expected$_{[VBN]}$ to$_{[TO]}$ race$_{[VB]}$ tomorrow$_{[NR]}$ .




$$P(VB|TO) = 0.83$$

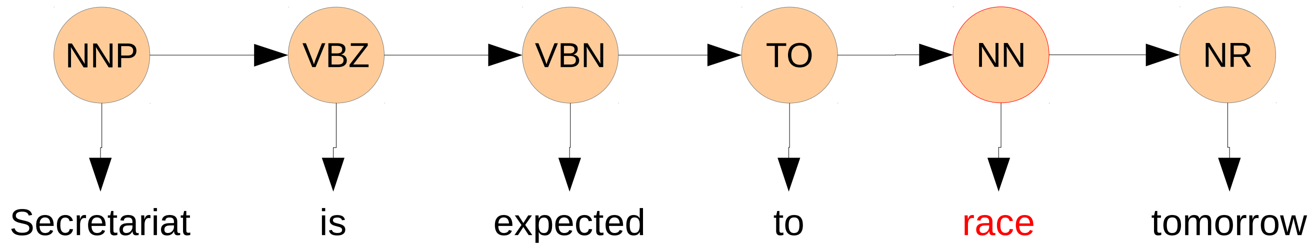
$$P(race|VB) = 0.00012$$

$$P(NR|VB) = 0.0027$$

$$P(VB|TO).P(NR|VB).P(race|VB) = 0.00000027$$

# Ambiguity

Secretariat$_{[NNP]}$ is$_{[VBZ]}$ expected$_{[VBN]}$ to$_{[TO]}$ race$_{[VB]}$ tomorrow$_{[NR]}$ .




P(NN|TO) = 0.00047

P(race|NN) = 0.00057

P(NR|NN) = 0.0012

P(NN|TO).P(NR|NN).P(race|NN) = 0.00000000032

# Viterbi algorithm

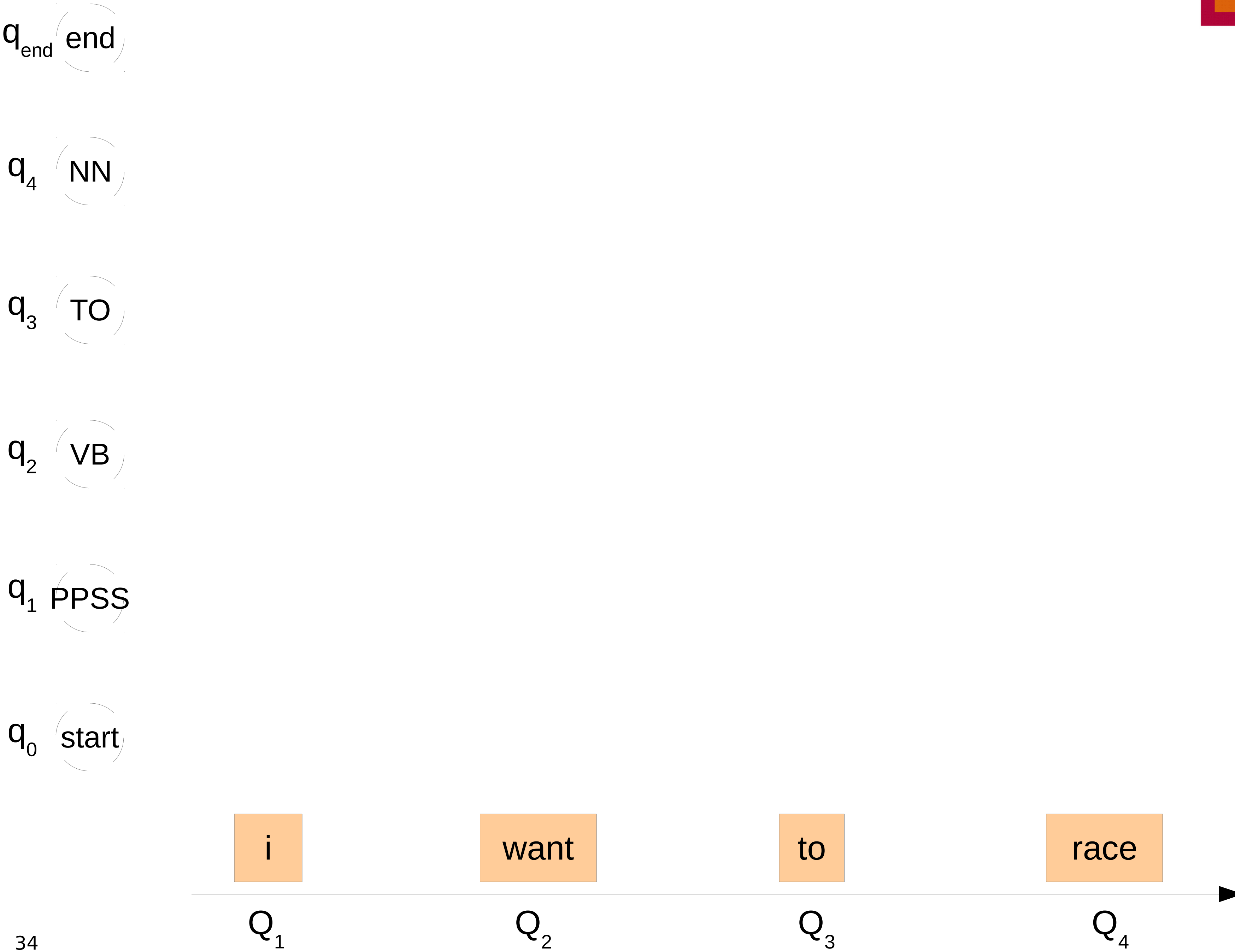
- Decoding algorithm for HMM
  - Determine the best sequence of POS tags

- Probability matrix
  - Columns corresponding to inputs (words)
  - Rows corresponding to possible states (POS tags)

# Viterbi algorithm

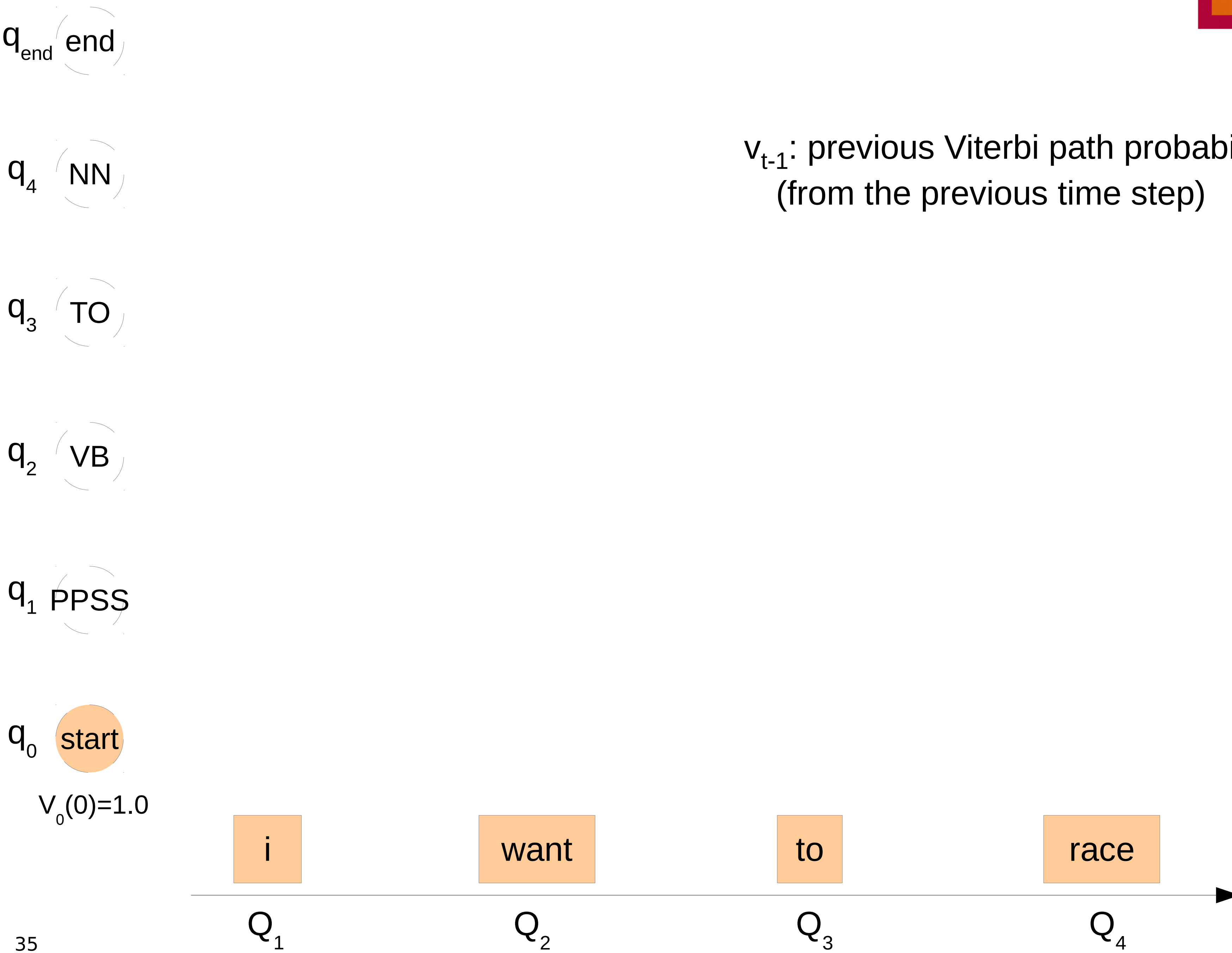
1. Move through the matrix in one pass filling the columns left to right using the transition probabilities and observation probabilities

2. Store the max probability path to each cell (not all paths) using dynamic programming



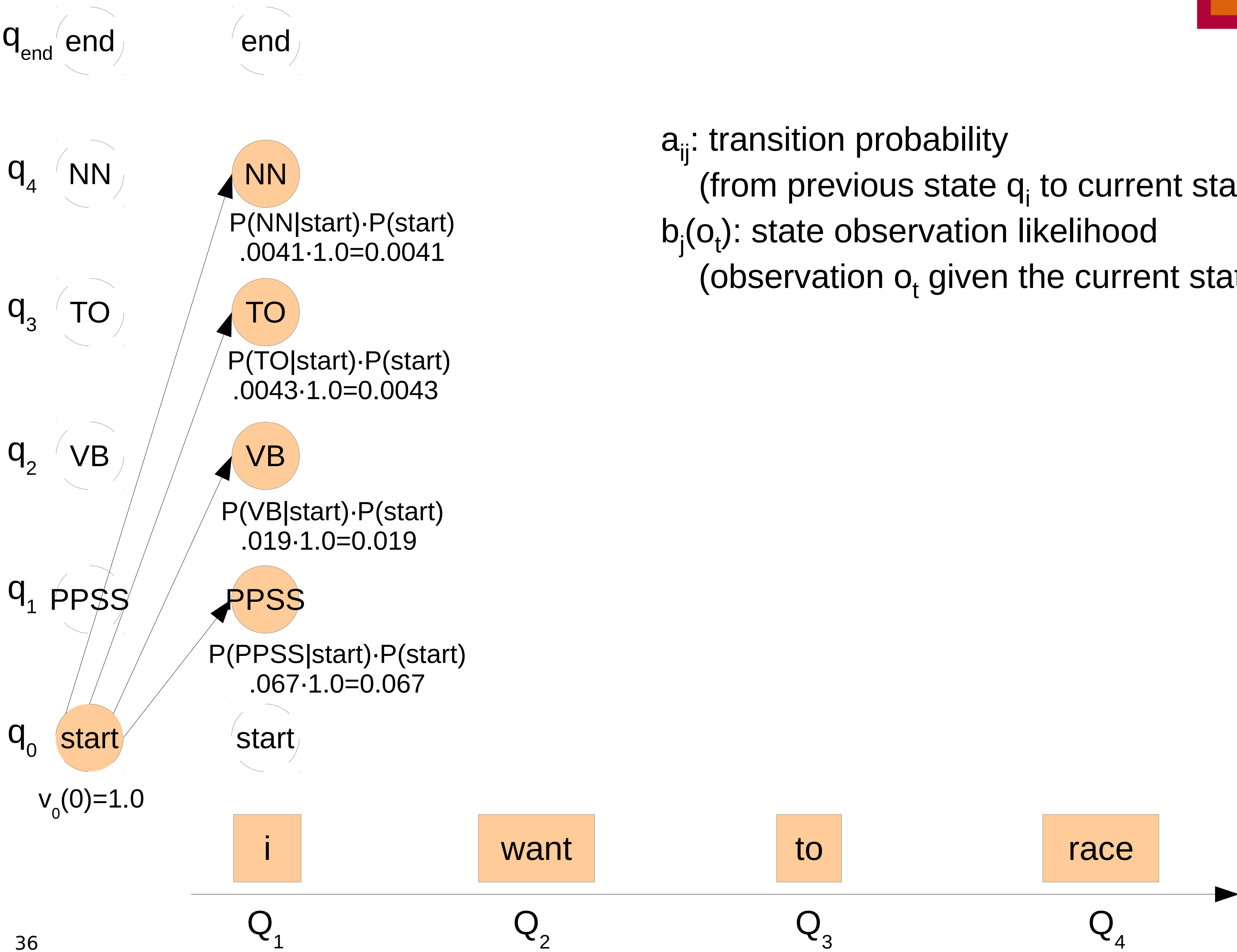
q_end end
q_4 NN
q_3 TO
q_2 VB
q_1 PPSS
q_0 start
i
want
to
race
Q_1
Q_2
Q_3
Q_4

$v_{t-1}$: previous Viterbi path probability
(from the previous time step)

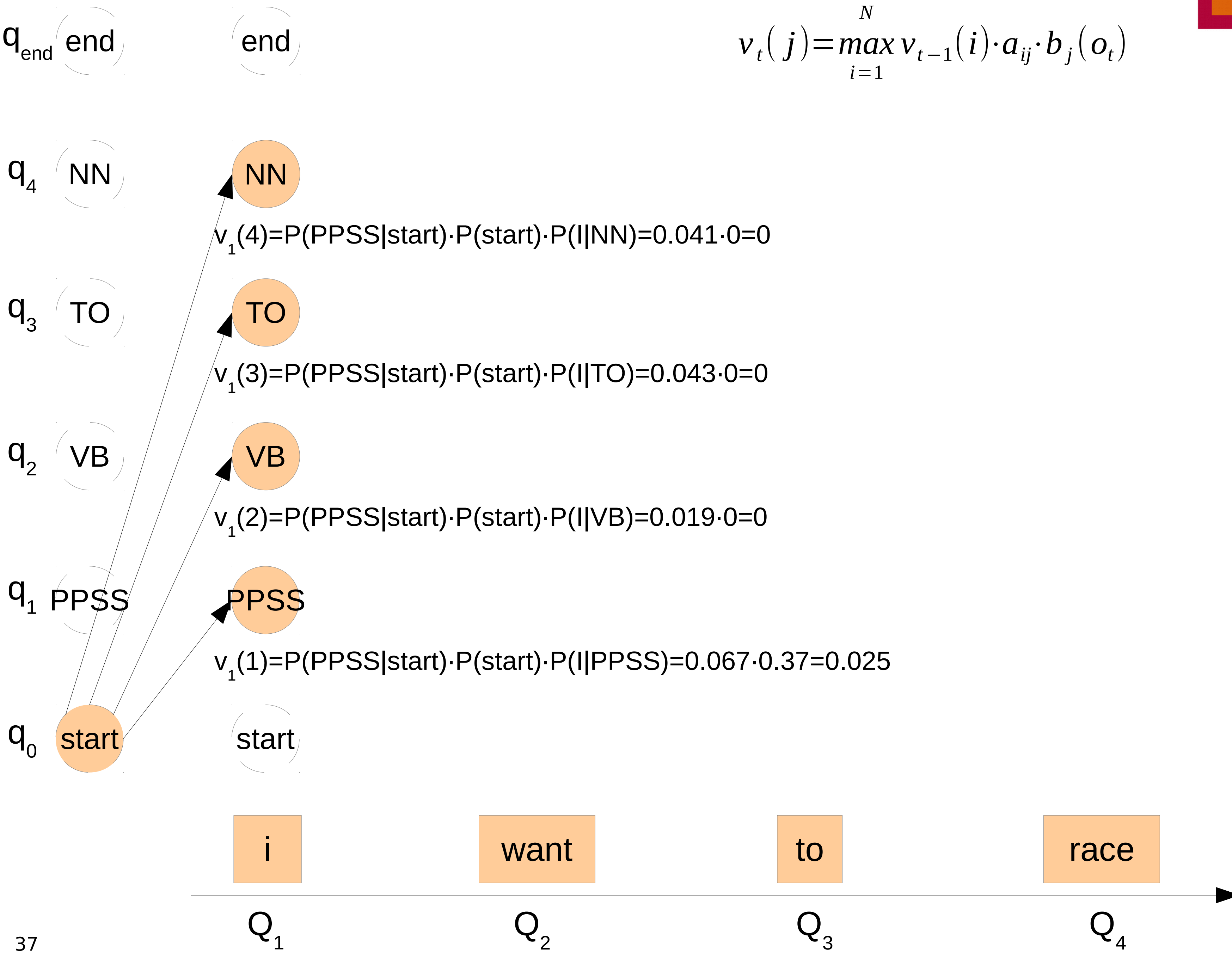


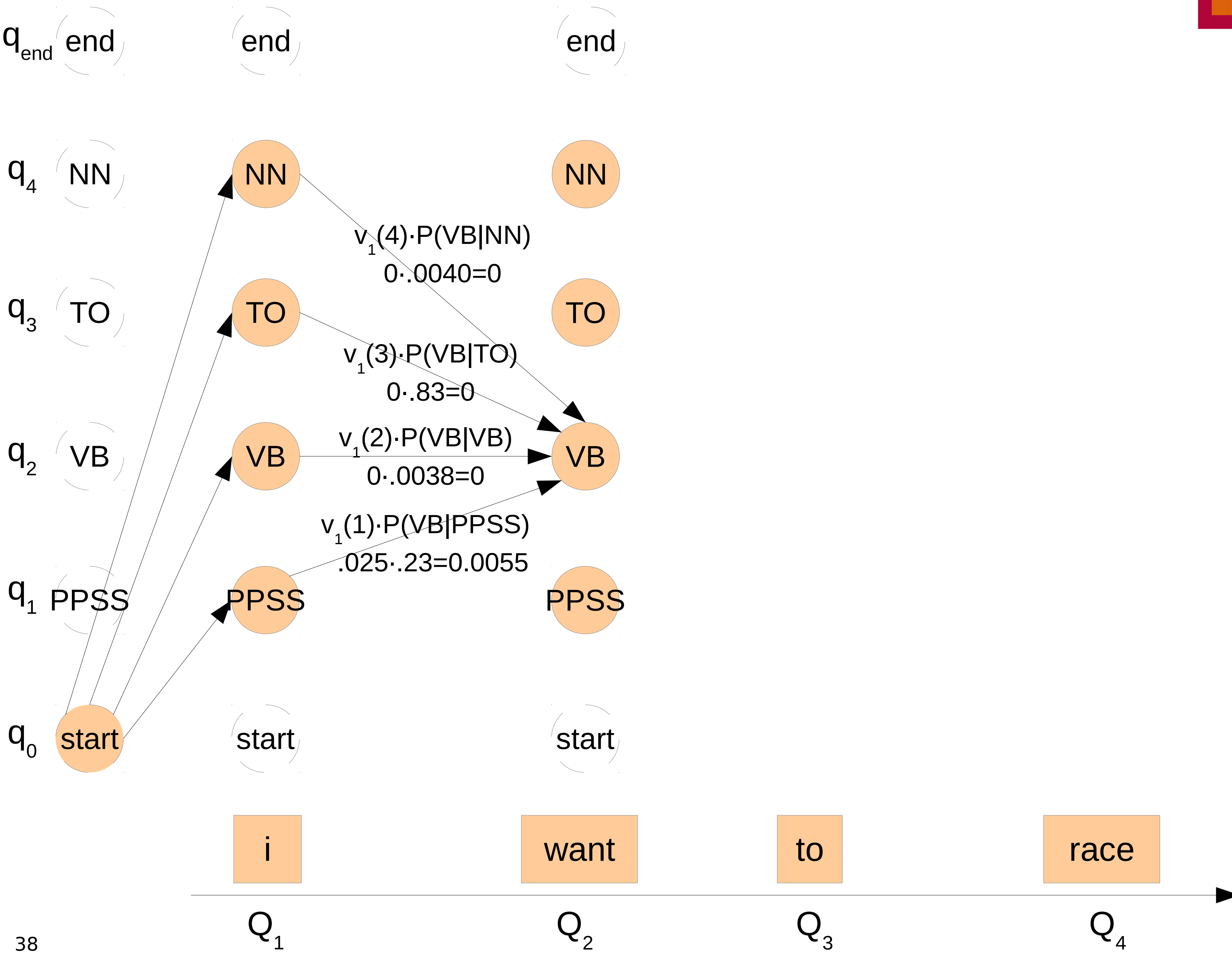
$a_{ij}$: transition probability
(from previous state $q_i$ to current state $q_j$)

$b_j(o_t)$: state observation likelihood
(observation $o_t$ given the current state j)

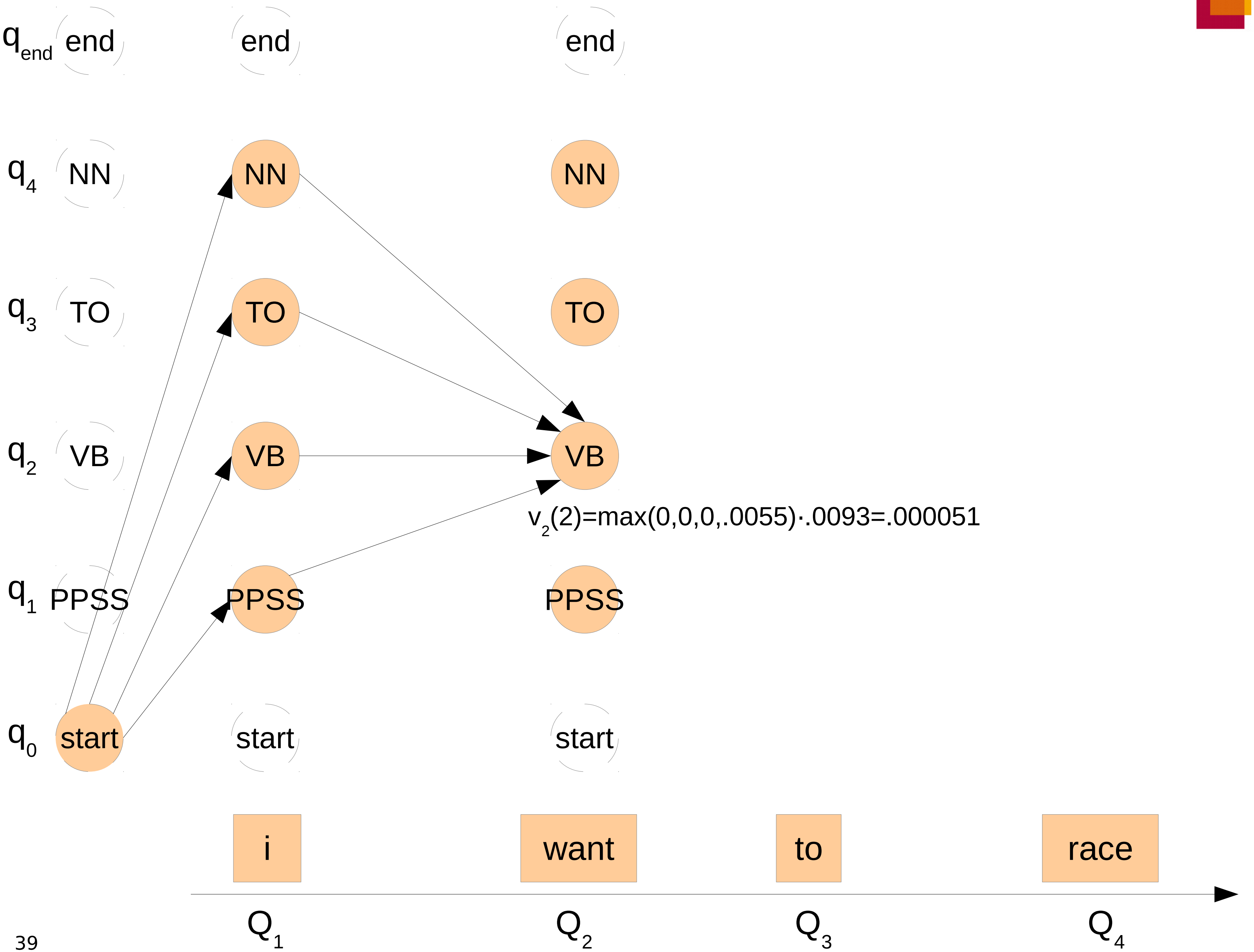


$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) \cdot a_{ij} \cdot b_j(o_t)$$

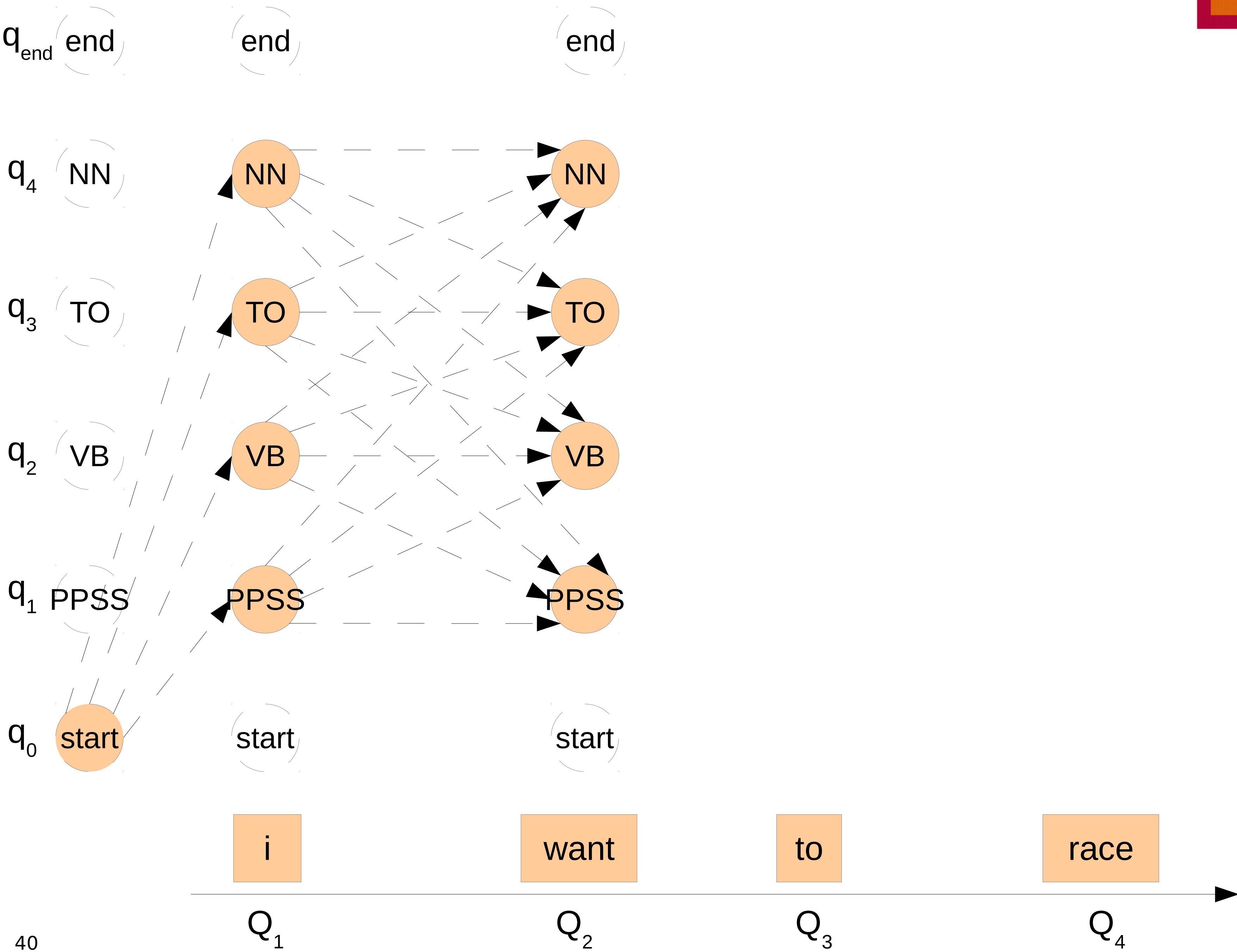




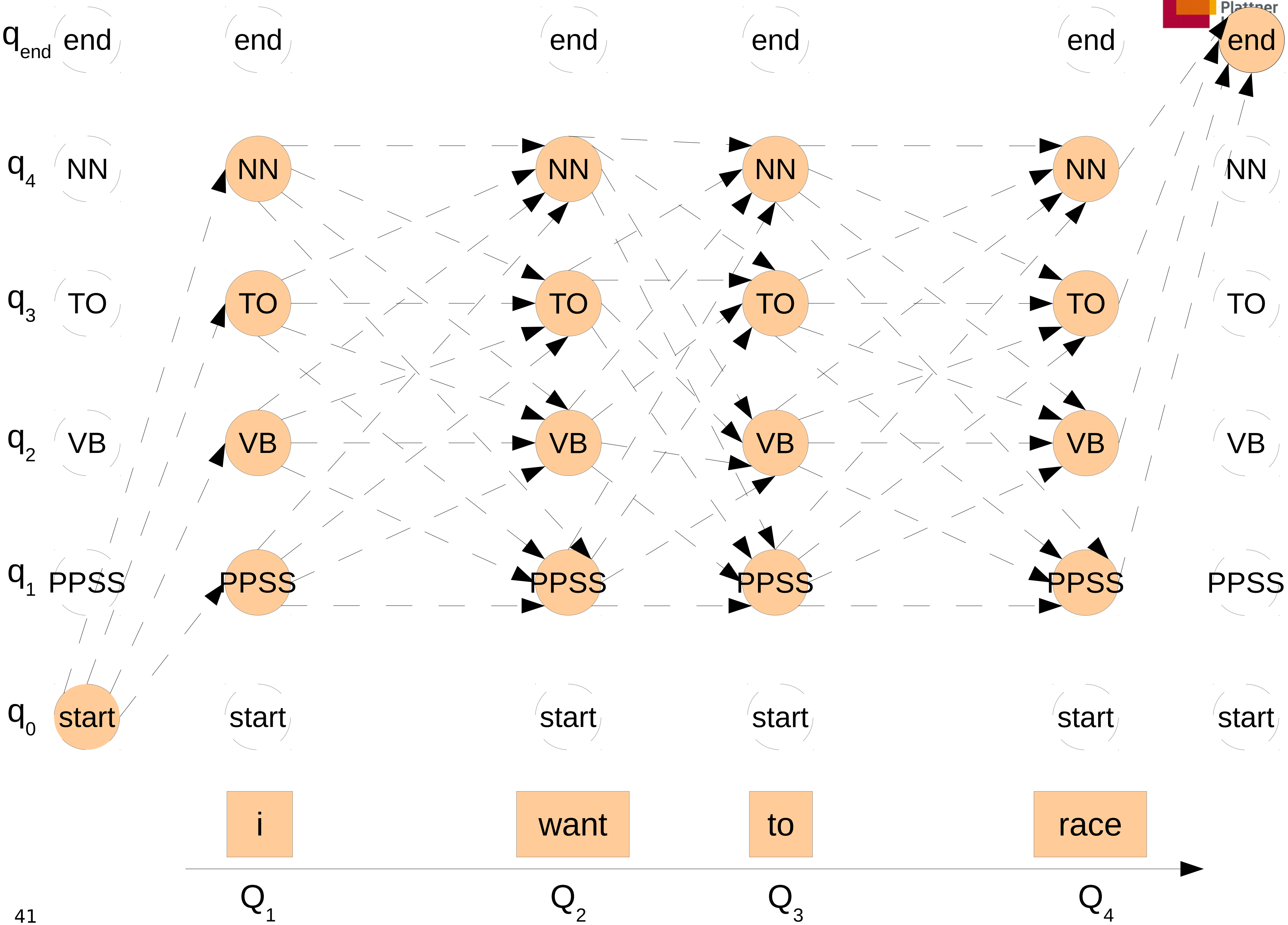
q_end
q_4
q_3
q_2
q_1
q_0
end
NN
TO
VB
PPSS
start
v_1(4)·P(VB|NN)
0·.0040=0
v_1(3)·P(VB|TO)
0·.83=0
v_1(2)·P(VB|VB)
0·.0038=0
v_1(1)·P(VB|PPSS)
.025·.23=0.0055
i
want
to
race
Q_1
Q_2
Q_3
Q_4

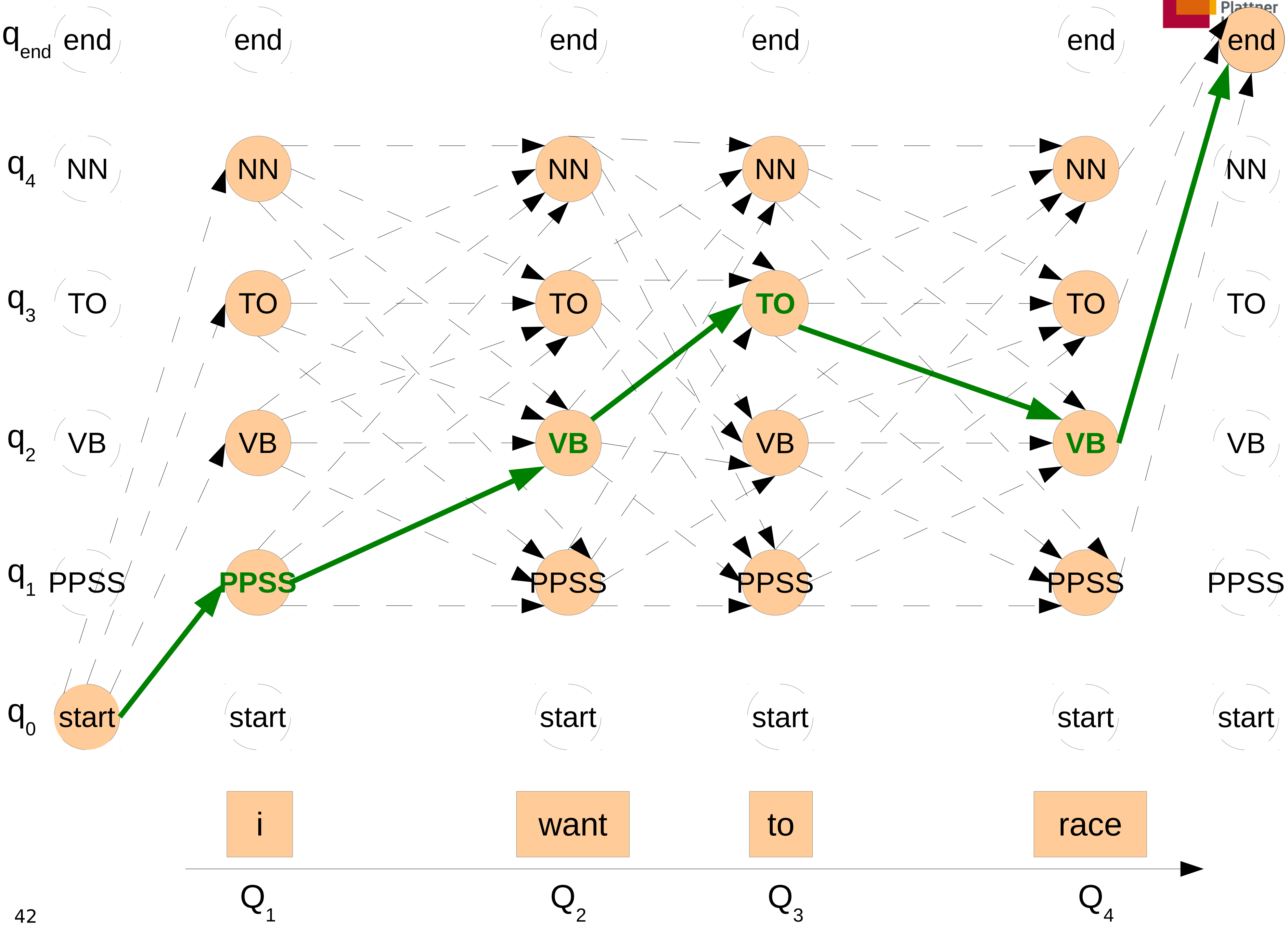


$q_{end}$
$q_4$
$q_3$
$q_2$
$q_1$
$q_0$
end
end
end
NN
NN
NN
TO
TO
TO
VB
VB
VB
PPSS
PPSS
PPSS
start
start
start
$v_2(2)=\max(0,0,0,.0055)\cdot.0093=.000051$
i
want
to
race
$Q_1$
$Q_2$
$Q_3$
$Q_4$



q end
end
end
end
q 4
NN
NN
NN
q 3
TO
TO
TO
q 2
VB
VB
VB
q 1
PPSS
PPSS
PPSS
q 0
start
start
start
i
want
to
race
Q 1
Q 2
Q 3
Q 4


q_end
q_4
q_3
q_2
q_1
q_0
end
NN
TO
VB
PPSS
start
i
want
to
race
Q_1
Q_2
Q_3
Q_4



$q_{end}$
$q_4$
$q_3$
$q_2$
$q_1$
$q_0$
end
NN
TO
VB
PPSS
start
i
want
to
race
$Q_1$
$Q_2$
$Q_3$
$Q_4$

# POS tagging using machine learning

- Classification problem (token by token) using a rich set of features

| | | |
|---|---|---|
| Current word | $w_i$ | $\& \; t_i$ |
| Previous word | $w_{i-1}$ | $\& \; t_i$ |
| Next word | $w_{i+1}$ | $\& \; t_i$ |
| Bigram features | $w_{i-1}, w_i$ | $\& \; t_i$ |
| | $w_i, w_{i+1}$ | $\& \; t_i$ |
| Previous tag | $t_{i-1}$ | $\& \; t_i$ |
| Tag two back | $t_{i-2}$ | $\& \; t_i$ |
| Next tag | $t_{i+1}$ | $\& \; t_i$ |
| Tag two ahead | $t_{i+2}$ | $\& \; t_i$ |
| Tag Bigrams | $t_{i-2}, t_{i-1}$ | $\& \; t_i$ |
| | $t_{i-1}, t_{i+1}$ | $\& \; t_i$ |
| | $t_{i+1}, t_{i+2}$ | $\& \; t_i$ |
| Tag Trigrams | $t_{i-2}, t_{i-1}, t_{i+1}$ | $\& \; t_i$ |
| | $t_{i-1}, t_{i+1}, t_{i+2}$ | $\& \; t_i$ |
| Tag 4-grams | $t_{i-2}, t_{i-1}, t_{i+1}, t_{i+2}$ | $\& \; t_i$ |
| Tag/Word combination | $t_{i-1}, w_i$ | $\& \; t_i$ |
| | $t_{i+1}, w_i$ | $\& \; t_i$ |
| | $t_{i-1}, t_{i+1}, w_i$ | $\& \; t_i$ |
| Prefix features | prefixes of $w_i$ (up to length 10) | $\& \; t_i$ |
| Suffix features | suffixes of $w_i$ (up to length 10) | $\& \; t_i$ |
| Lexical features | whether $w_i$ has a hyphen | $\& \; t_i$ |
| | whether $w_i$ has a number | $\& \; t_i$ |
| | whether $w_i$ has a capital letter | $\& \; t_i$ |
| | whether $w_i$ is all capital | $\& \; t_i$ |

(https://link.springer.com/chapter/10.1007/11573036_36)

# POS tagging using neural networks

- e.g., using Bidirectional Long Short-Term Memory Recurrent Neural Network (bi-LSTM)
- Input based on tokens, characters and bytes

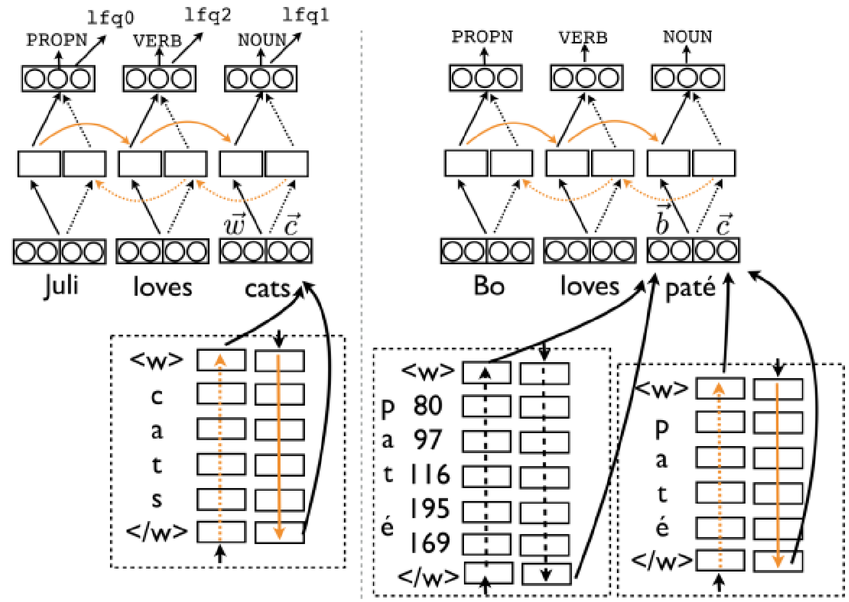



Figure 1: Right: bi-LSTM, illustrated with $\vec{b} + \vec{c}$ (bytes and characters), for $\vec{w} + \vec{c}$ replace $\vec{b}$ with words $\vec{w}$. Left: FREQBIN, our multi-task bi-LSTM that predicts at every time step the tag and the frequency class for the next token.

 (https://www.aclweb.org/anthology/P/P16/P16-2067.pdf)

# Evaluation

- Corpus
  - Training and test, and optionally also development set
  - Training (cross-validation) and test set

- Evaluation
  - Comparison of gold standard (GS) and predicted tags
  - Evaluation in terms of Precision, Recall and F-Measure

# Precision and Recall

- Precision:
  - Amount of labeled items which are correct

$$Precision = \frac{tp}{tp + fp}$$

- Recall:
  - Amount of correct items which have been labeled

$$Recall = \frac{tp}{tp + fn}$$

# F-Measure

- There is a strong anti-correlation between precision and recall
- Having a trade off between these two metrics
- Using F-measure to consider both metrics together
- F -measure is a weighted harmonic mean of precision and recall

$$F = \frac{(\beta^2 + 1) P R}{\beta^2 P + R}$$

# Error Analysis

- Confusion matrix or contingency table
  - Percentage of overall tagging error

| | **IN** | **JJ** | **NN** | **NNP** | **RB** | **VBD** | **VBN** |
|---|---|---|---|---|---|---|---|
| **IN** | - | .2 | | | .7 | | |
| **JJ** | .2 | - | 3.3 | 2.1 | 1.7 | .2 | 2.7 |
| **NN** | | 8.7 | - | | | | .2 |
| **NNP** | .2 | 3.3 | 4.1 | - | .2 | | |
| **RB** | 2.2 | 2.0 | .5 | | - | | |
| **VBD** | | .3 | .5 | | | - | 4.4 |
| **VBN** | | 2.8 | | | | 2.6 | |

# Summary

- POS tagging and tagsets
- Rule-based algorithms
- Sequential algorithms
- Neural networks
- Evaluation (P,R,FM)

# Tools for POS tagging

- Spacy: https://spacy.io/
- OpenNLP: https://opennlp.apache.org/
- Stanford CoreNLP: https://stanfordnlp.github.io/CoreNLP/
- NLTK Python: http://www.nltk.org/
- and others...

# Further reading

- Book Jurafski & Martin
  - Chapter 5

# Exercise

- Project: choose a POS tagger and use it in your project.
  - Can POS tags support your task?