# NLP and the Web – WS 2024/2025

Menti.com
7544 5229

# Lecture 2
# Foundations of Text Classification

**Dr. Thomas Arnold**
**Hovhannes Tamoyan**
**Kexin Wang**

**Ubiquitous Knowledge Processing Lab**
**Technische Universität Darmstadt**

# Syllabus (tentative)

Menti.com
7544 5229

| Nr. | Lecture |
|---|---|
| 01 | Introduction / NLP basics |
| **02** | **Foundations of Text Classification** |
| 03 | IR – Introduction, Evaluation |
| 04 | IR – Word Representation, Data Collection |
| 05 | IR – Re-Ranking Methods |
| 06 | IR – Language Domain Shifts, Dense / Sparse Retrieval |
| 07 | LLM – Language Modeling Foundations |
| 08 | LLM – Neural LLM, Tokenization |
| 09 | LLM – Transformers, Self-Attention |
| 10 | LLM – Adaption, LoRa, Prompting |
| 11 | LLM – Alignment, Instruction Tuning |
| 12 | LLM – Long Contexts, RAG |
| 13 | LLM – Scaling, Computation Cost |
| 14 | Review & Preparation for the Exam |

# Last Lecture – Linguistic Analysis Levels

Menti.com
7544 5229

- Phonetics and Phonology
- Segmentation
- Morphology
- Syntax
- Semantics
- Pragmatics and Discourse

# Today's lecture

Menti.com
7544 5229

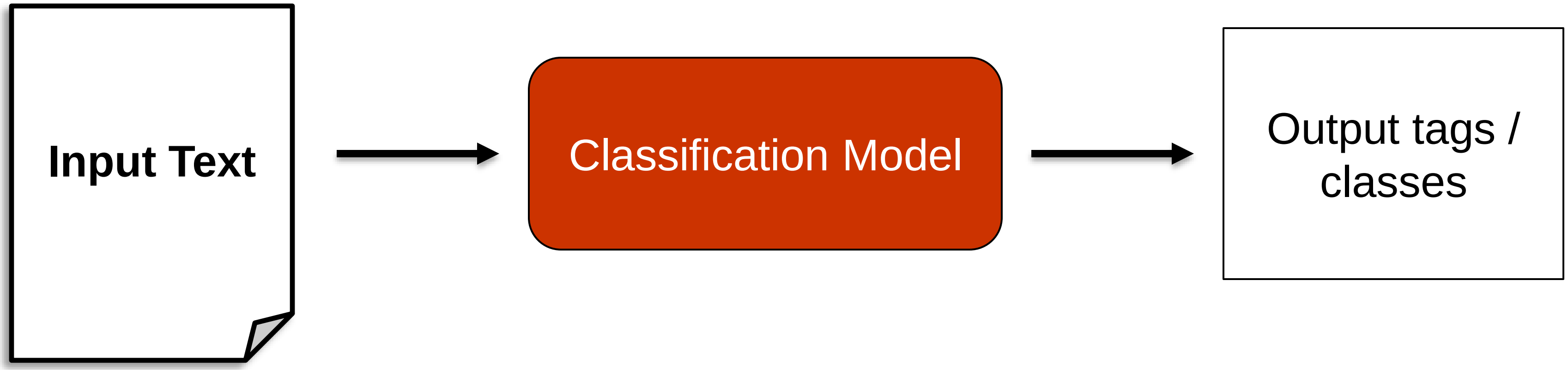
**Text Classification: Introduction**

**Algorithms**

- **Naive Bayes**
- **Hidden Markov Models**

# What is Text Classification?

Menti.com
7544 5229



# Examples – Spam Detection

Menti.com
7544 5229

From: Danke (**archdigest@news.condenast.com)**

Sehr geehrter kunde,

Seit Sie unsere Dienste nutzen, haben Sie **40,259** Treuepunkte gesammelt. Sie erhalten ein Handy als Geschenk für Ihre Treue.

Der Versand erfolgt nach Bestätigung Ihrer Anschrift und Bezahlung der Versandkosten

Dein Geschenk: **Apple iPhone X (64GB) - Space Grey**

* Indem Sie Ihren Gewinn akzeptieren, stimmen Sie zu, dass Ihr Konto mit 35.154 Punkten belastet wird. Ohne Ihre vorherige Zustimmung wird kein Abonnement abonniert.

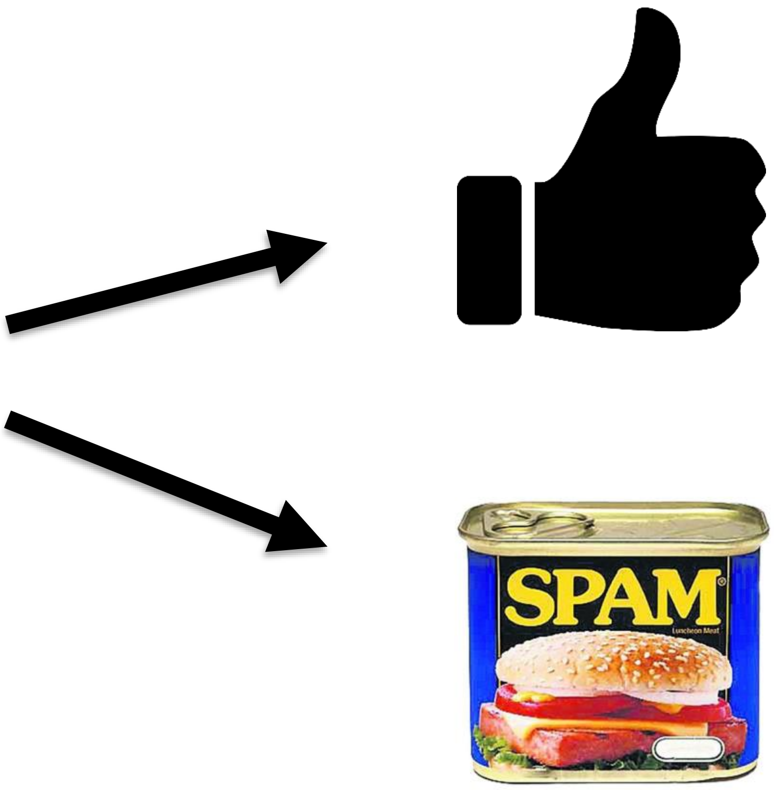
Ihre Treuepunkte verfallen bald. Fordern Sie Ihre Prämien vor Ablauf des Angebots an.

Klicken Sie auf die Schaltfläche unten, um zu bestätigen und Ihr Handy zu erhalten.

KLICK HIER

Grüße,

Julien I Produktmanager



# Examples – Sentiment Analysis

Menti.com
7544 5229

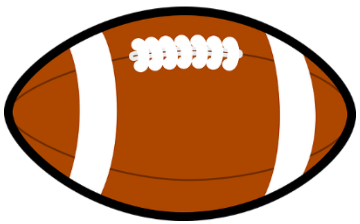
Positive or negative movie review?

- This is it. This is the one. This is the worst movie ever made. Ever. It beats everything. I have never seen worse.
- Expertly scripted and perfectly delivered, this searing parody leaves you literally rolling with laughter.
- While watching this film I started to come up with things I would rather be doing, including drinking bleach, rubbing sand in my eyes, and tax returns.
- Just finished watching this movie for maybe the 7th or 8th time

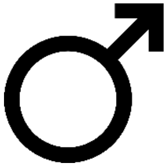
# More Examples

Menti.com
7544 5229

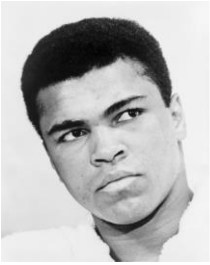
Topic Labeling



Age/Gender Identification



**I've seen George Foreman shadow boxing and the shadow won.**



Authorship Identification

Language Identification

# Approaches
# Rule-based

Menti.com
7544 5229

Handcrafted linguistic rules

Human comprehensible

Pro: Precision can be high

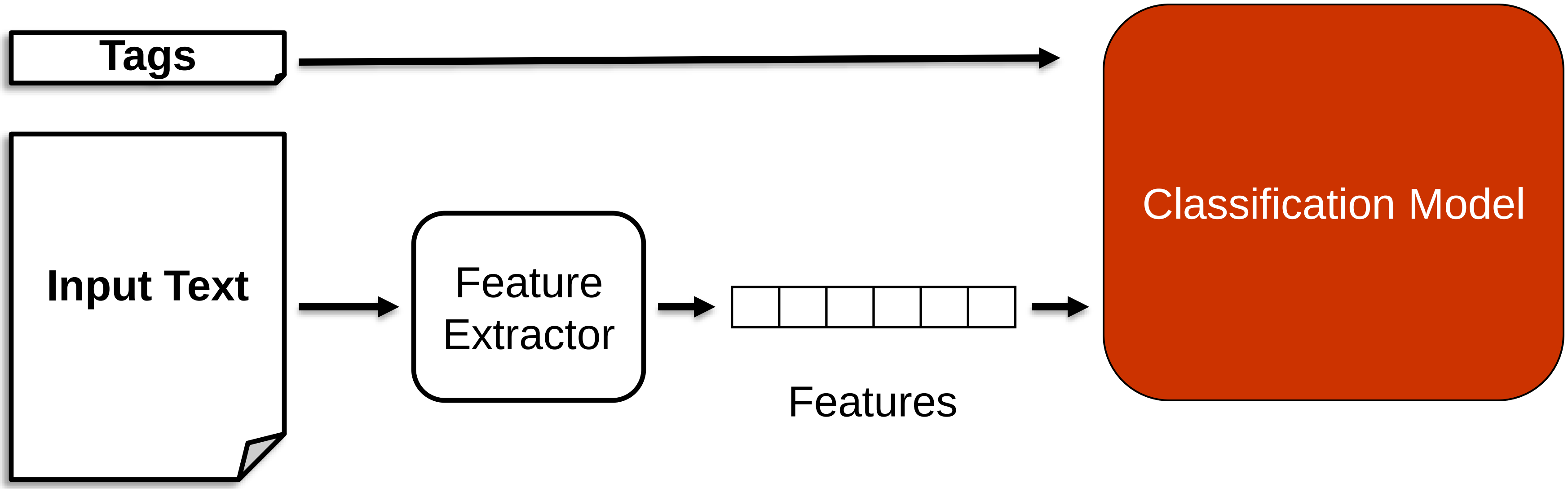
Con: Very expensive to build and maintain

```
IF "basketball" THEN
            return top_sports
ELSEIF…
```

# Approaches
# Supervised Machine Learning

Menti.com
7544 5229

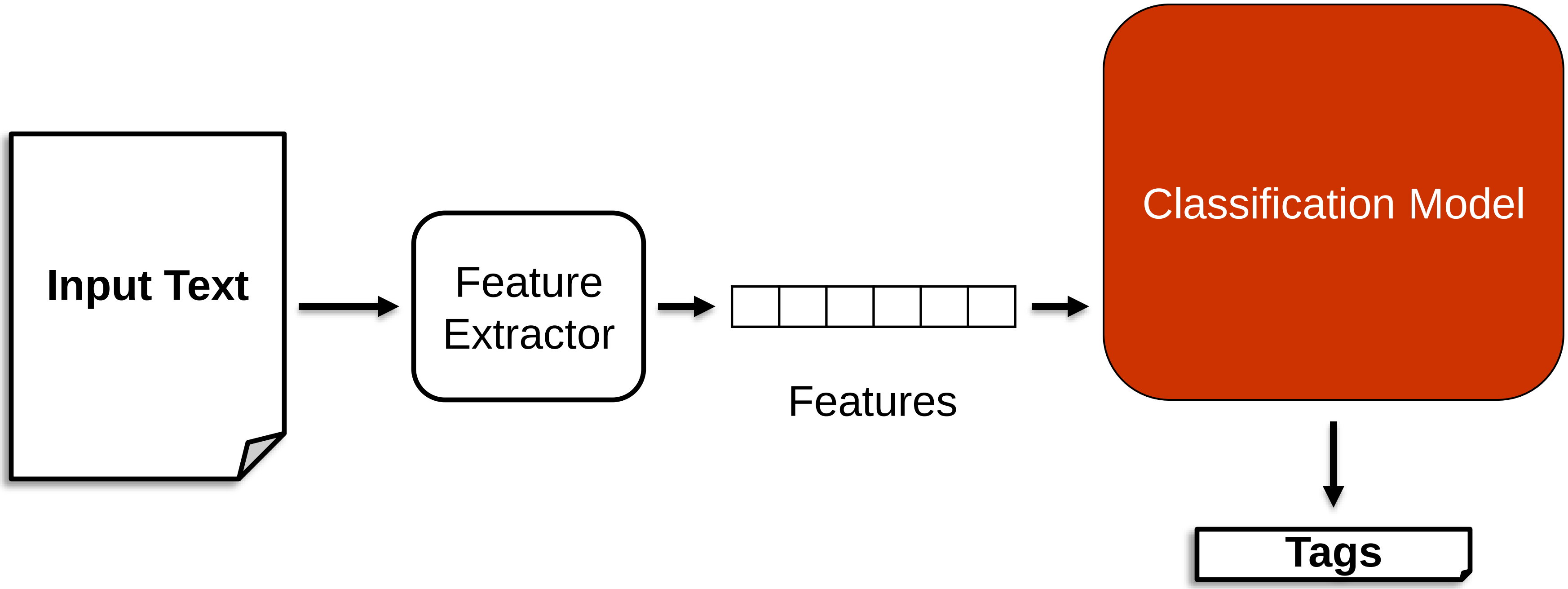
## Step 1: Training



# Approaches
# Supervised Machine Learning

Menti.com
7544 5229

## Step 2: Prediction



# Approaches
# Supervised Machine Learning

Menti.com
7544 5229

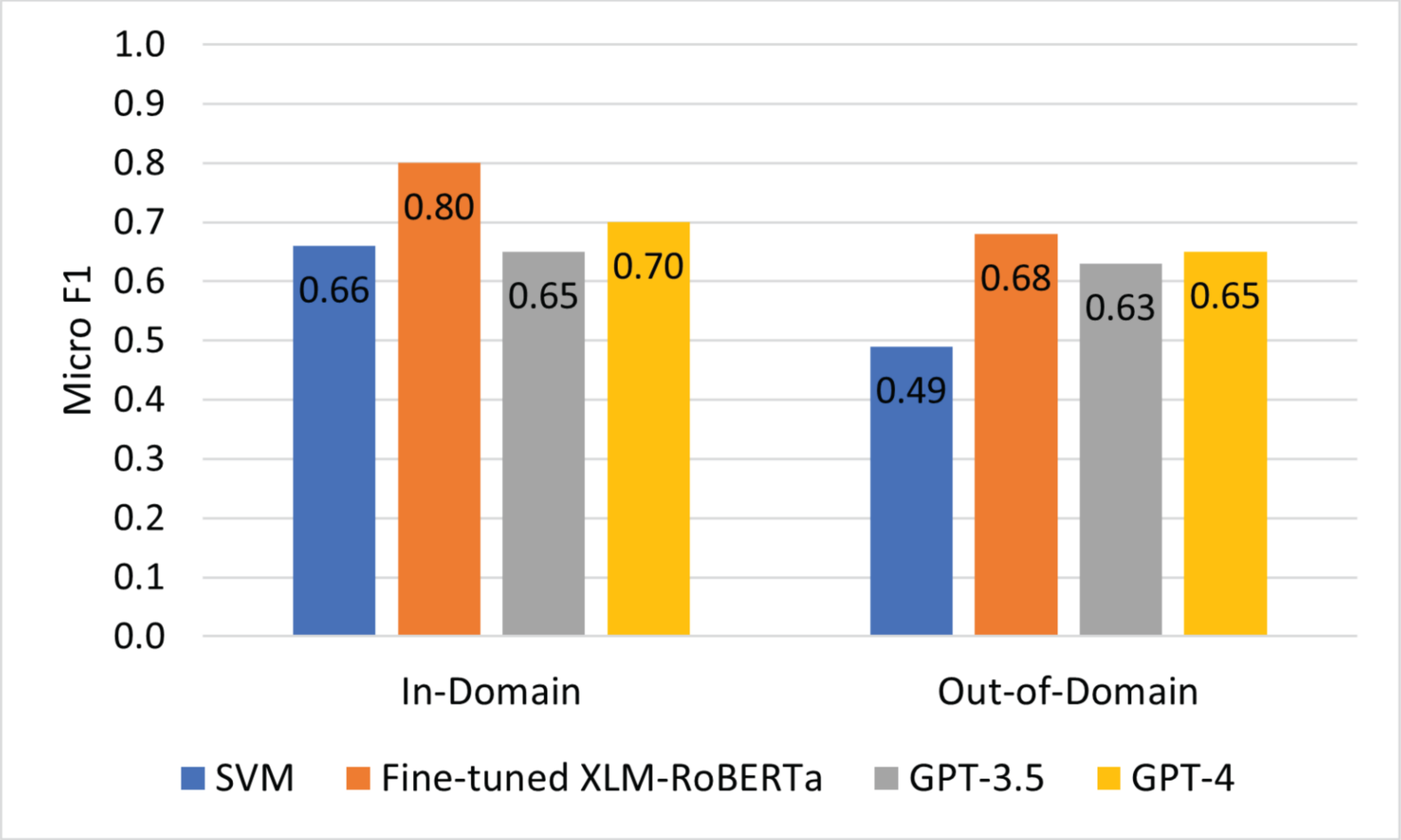
Classification model based on Training Data

Human comprehensible? Dependant on model!

Pro: Easier to maintain, usually more accurate

Con: Needs training data

# Comparison to Neural Models


Kuzman T, Mozetič I, Ljubešić N. Automatic Genre Identification for Robust Enrichment of Massive Text Collections: Investigation of Classification Methods in the Era of Large Language Models. Machine Learning and Knowledge Extraction. 2023; 5(3):1149-1175

# Today's lecture

Menti.com
7544 5229

**Text Classification: Introduction**

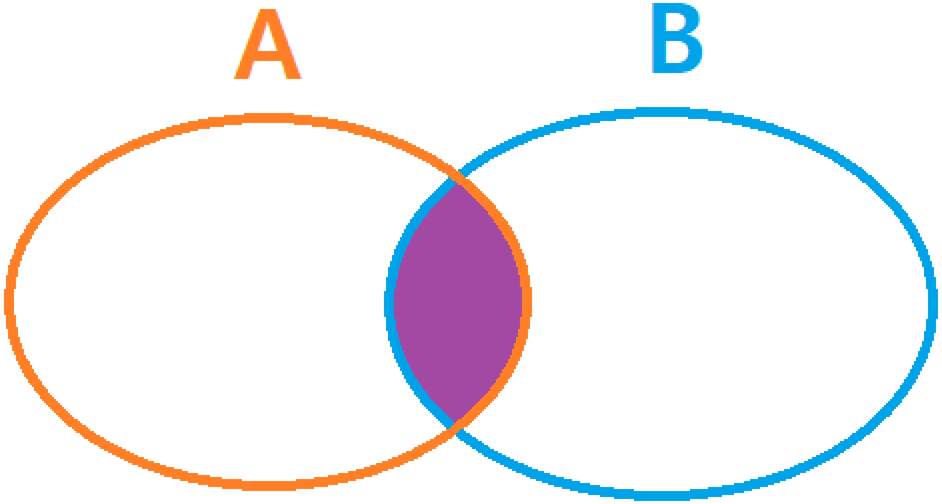
**Algorithms**

- **Naive Bayes**
- **Hidden Markov Models**

# Naïve Bayes Classifier: Background

Menti.com
7544 5229

- Before starting let's review related concepts:
  - Conditional Probability
  - Bayes' Rule.



# Conditional Probability

Menti.com
7544 5229

- **Conditional Probability** in plain English:
  - What is the probability that something will happen, *given that something else* has already happened.
- Assume we have some **Outcome** O and some **Evidence** E.
  - $P(O, E)$: the Probability of having *both* the Outcome O and Evidence E is the multiplication of two probabilities:
    - $P(O)$: Probability of O occurring
    - $P(E|O)$: Probability of E given that O happened

$$P(O, E) = P(O) \times P(E|O)$$

# Conditional Probability: Example

Menti.com
7544 5229

- Let say we have a group of students
  - Students could be either awake or asleep
  - Students are also either bachelor or master students
- If we select one student randomly: *what is the probability that this person is a sleeping bachelor student?*
- Conditional Probability can help us answer that:
  - $P(Asleep\,, Bachelor) = P(Asleep) * P(Bachelor|Asleep)$
- We could compute the exact same thing, the reverse way
  - $P(Asleep\,, Bachelor) = P(Bachelor) * P(Asleep\,|Bachelor)$

# Bayes Rule

Menti.com
7544 5229

- Conceptually, this is a way to go from $P(Evidence \mid Outcome)$ to $P(Outcome \mid Evidence)$
- Let's see how to do that
  - We have:
    - $P(O,E) = P(O) \times P(E|O)$ and $P(O,E) = P(E) \times P(O|E)$
    - $\rightarrow P(O) \times P(E|O) = P(E) \times P(O|E)$

$$P(O|E) = \frac{P(E|O) \times P(O)}{P(E)}$$

Bayes Rule

# Getting to Naïve Bayes'

Menti.com
7544 5229

- So far, we have talked only about one piece of evidence.
- In reality, we have to predict an outcome given **multiple evidence**
  - math gets very complicated :(
- Naive Bayes‘ solution:
  - treat each of piece of evidence as independent -> simpler math :)
  - This approach is why this is called ***naive*** Bayes
- Suppose we have multiple evidences $E_1, \dots, E_n$ and an outcome $O$

$$P(O|E_1, \dots, E_n) = \frac{P(E_1|O) \times P(E_2|O) \times \dots \times P(E_n|O) \times P(O)}{P(E_1, E_2, \dots, E_n)}$$

# Getting to Naïve Bayes'

Menti.com
7544 5229

$$P(O|E_1, \ldots, E_n) = \frac{P(E_1|O) \times \ P(E_2|O) \times \ \ldots \times \ P(E_n|O) \times P(O)}{P(E_1, E_2, \ldots, E_n)}$$

Many people choose to remember this as:

$$P(outcome|evidence) = \frac{P(Likelihood\ of\ Evidence) \ * \ Prior\ prob\ of\ outcome}{P(Evidence)}$$

- Notes:
- If the $P(evidence|outcome)$ is 1, then we are just multiplying by 1.
- If the $P(some\ particular\ evidence|outcome)$ is 0, then the whole probability becomes 0
- Since we divide everything by $P(Evidence)$,
  - we can even get away without calculating it.
- The intuition behind multiplying by the *prior* is
  - to give high probability to more common outcomes, and low probabilities to unlikely outcomes.
  - These are also called **base rates** and they are a way to scale our predicted probabilities

# Naïve Bayes: Example

Menti.com
7544 5229

- Let's say that we have data on 1000 pieces of fruit: **Banana**, **Orange** or some **Other Fruit**
- We know 3 characteristics about each fruit
  - Whether it is **Long**
  - Whether it is **Sweet** and
  - If its color is **Yellow**
- Our training set

| Type | Long | Not Long | Sweet | Not Sweet | Yellow | Not Yellow | Total |
|---|---|---|---|---|---|---|---|
| Banana | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| Orange | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| Other Fruit | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

# Naïve Bayes: Example cont.

Menti.com
7544 5229

| Type | Long | Not Long | Sweet | Not Sweet | Yellow | Not Yellow | Total |
|---|---|---|---|---|---|---|---|
| Banana | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| Orange | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| Other Fruit | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

- “Prior” probabilities
  - P(Banana) = 500/1000 = 0.5, P(Orange) = 0.3, P(Other Fruit) = 0.2
- Probability of "Evidence“
  - p(Long) = 500/1000 = 0.5, P(Sweet) = 0.65, P(Yellow) = 0.8
- Probability of "Likelihood“
  - P(Long|Banana) = 0.8, P(Long|Orange) = 0
  - P(Yellow|Other Fruit) = 50/200 = 0.25, P(Not Yellow|Other Fruit) = 0.75

# Naïve Bayes: Example cont.

Menti.com
7544 5229

- "Prior" probabilities
  - P(Banana) = 0.5 (500/1000), P(Orange) = 0.3, P(Other Fruit) = 0.2
- Probability of "Evidence"
  - p(Long) = 500/100 = 0.5, P(Sweet) = 0.65, P(Yellow) = 0.8
- Probability of "Likelihood"
  - P(Long|Banana) = 0.8, P(Long|Orange) = 0
  - P(Yellow|Other Fruit) = 50/200 = 0.25, P(Not Yellow|Other Fruit) = 0.75

Given an unknown fruit which is long, sweet and yellow, is it Banana, Orange or Other Fruit?

- $\mathrm{P(Banana|Long, Sweet\ and\ Yellow)} = \frac{\mathrm{P(Long|Banana)} * \mathrm{P(Sweet|Banana)} * \mathrm{P(Yellow|Banana)} * \mathrm{P(Banana)}}{\mathrm{P(Long)} * \mathrm{P(Sweet)} * \mathrm{P(Yellow)}}$
  $= 0.8 * 0.7 * 0.9 * 0.5 / \mathrm{P(evidence)}$= 0.252 / P(evidence)
- $\mathrm{P(Orange|Long, Sweet\ and\ Yellow)} = 0$ , why?
- $\mathrm{P(Other\ Fruit|Long, Sweet\ and\ Yellow)} = \frac{\mathrm{P(Long|Other\ Fruit)} * \mathrm{P(Sweet|Other\ Fruit)} * \mathrm{P(Yellow|Other\ Fruit)} * \mathrm{P(Other\ Fruit)}}{\mathrm{P(Long)} * \mathrm{P(Sweet)} * \mathrm{P(Yellow)}}$
  $= (100/200 * 150/200 * 50/150 * 200/1000) \,/\, P(evidence) = 0.01875 \,/\, P(evidence)$

0.252 >> 0.01875 => the unknown fruit is most likely a banana

Is it clear now why we don‘t need to calculated the P(evidence)?

# Naïve Bayes for Text Classification based on Words as Features

Menti.com
7544 5229

- Multinomial Naïve Bayes Model: The probability of document *d* belonging to a class *c* is proportional to the product of the probabilities of terms *t* belonging to a class *c*, and to the class prior *P(c)*:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

- The best class *c* for a document *d* is found by selecting the class, for which the maximum a posteriori (map) probability is maximal:

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c).$$

# Summary on Naïve Bayes

Menti.com
7544 5229

- Bayesian methods provide the basis for probabilistic learning methods
- Bayesian methods can be used to determine the most probable hypothesis given the data
- No training, just probability calculation
- Binary, numeric and nominal features can be mixed
- Naïve Bayes fails if the independence assumption is violated too much
  - Especially identical or highly overlapping features pose a problem that has to be addressed with proper feature selection

# I didn't expect a kind of menti quiz.

Menti.com
7544 5229

# Today's lecture

Menti.com
7544 5229

**Text Classification: Introduction**

**Algorithms**

- **Naive Bayes**
- **Hidden Markov Models**

# Limitations of Standard classification

Menti.com
7544 5229

- Standard classification problem assumes
  - individual cases are disconnected and independent
- Many NLP problems do not satisfy this assumption
  - involve making many connected decisions, each resolving a different ambiguity, but which are mutually dependent
- More sophisticated learning and inference techniques are needed

# Sequence Labeling

Menti.com
7544 5229

- Many NLP problems can viewed as sequence labeling
- Each token in a sequence is assigned a label
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors
- Examples:
  - Part Of Speech Tagging

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| In: | John | saw | the | saw | and | decided | to | take | it | to | the | table. |
| Out: | PN | V | Det | N | Con | V | Part | V | Pro | Prep | Det | N |

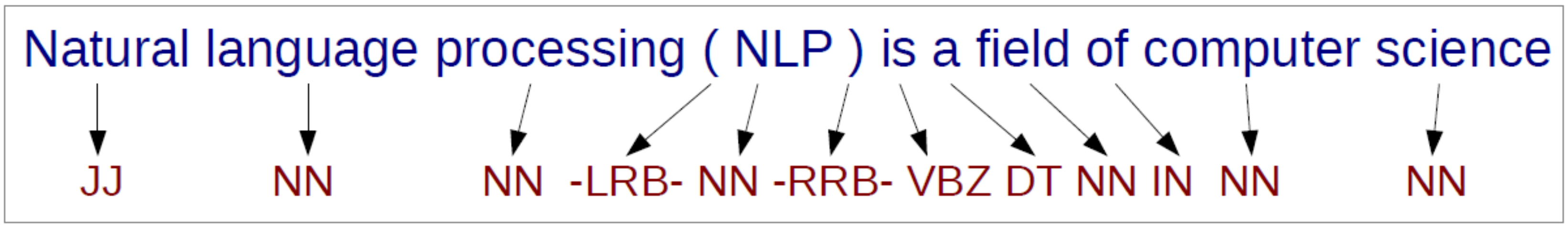
  - Named entity recognition: people organizations places

Michael Dell is the CEO of Dell Computer Corporation and lives in Austin Texas.

# Use case: Part of Speech (POS) Tagging

Menti.com
7544 5229

- Given a sentence X, predict its part of speech sequence Y




- A type of “structured” prediction
- How can we do this?

# Use case: Part of Speech (POS) Tagging

Menti.com
7544 5229

**Possible Answers**

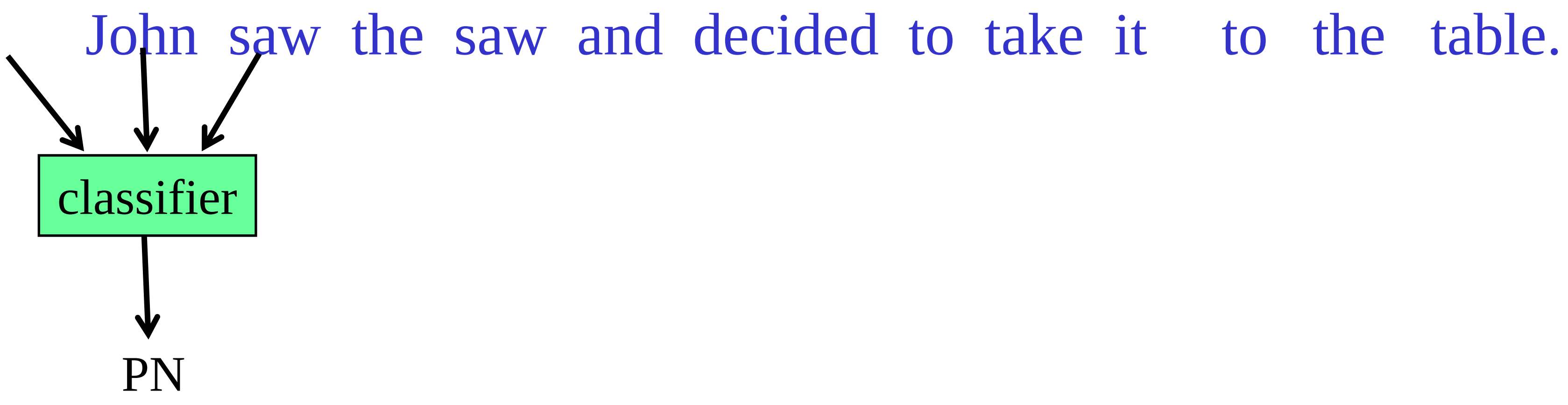
- Sequence labeling as classification:
  - Pointwise prediction: predict each word individually with a classifier
- Generative sequence models: e.g. Hidden Markov Model (HMM)

- Later in the lecture: Neural Sequence Models (RNN / LSTM)

# Sequence Labeling as Classification



- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

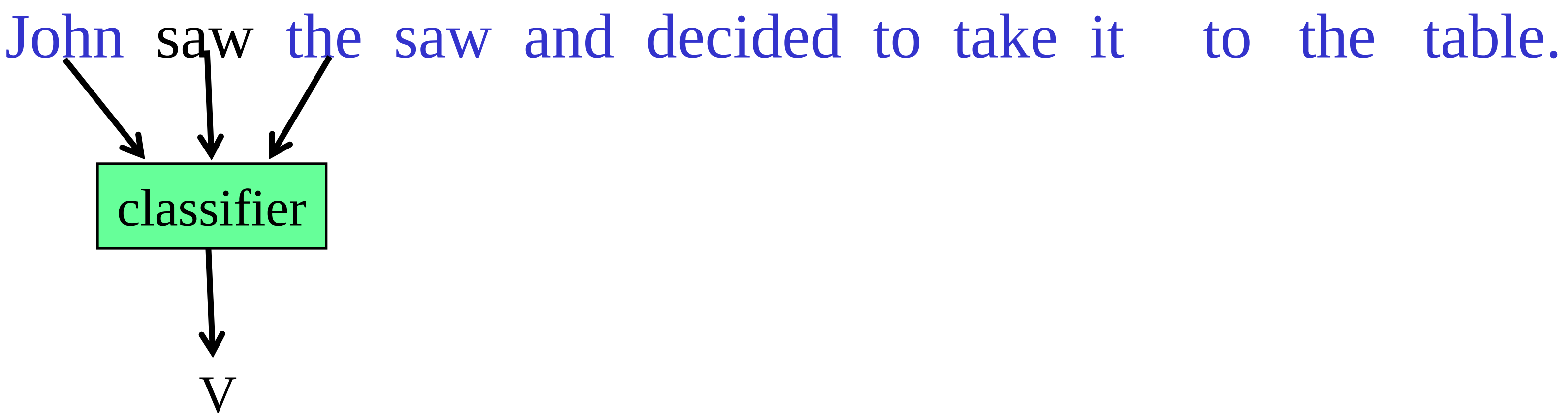



Slides from: Raymond J. Mooney

# Sequence Labeling as Classification

Menti.com
7544 5229

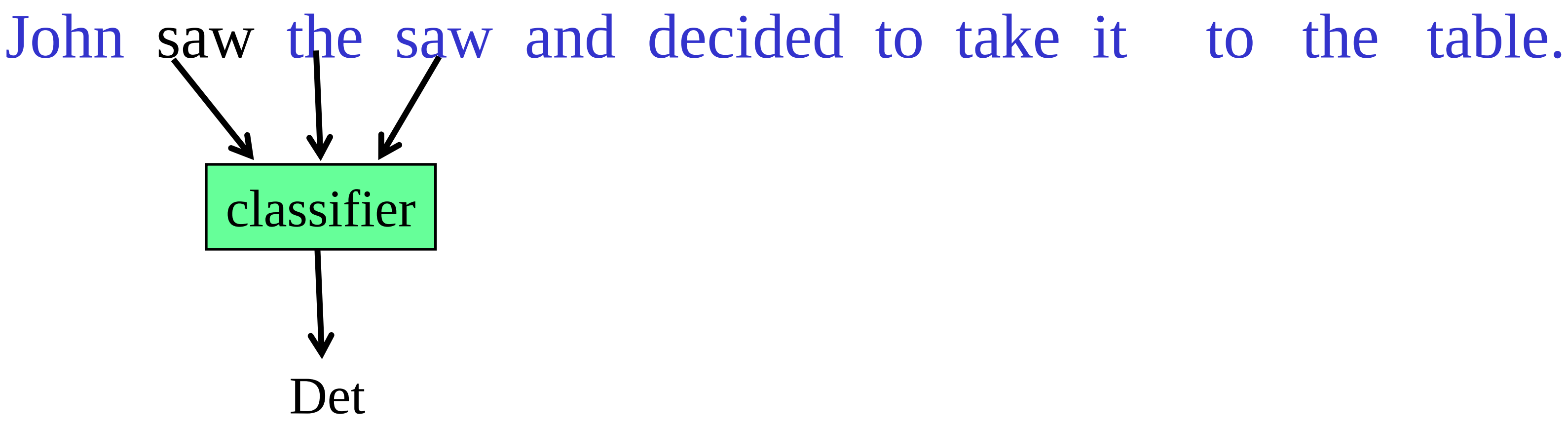
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification



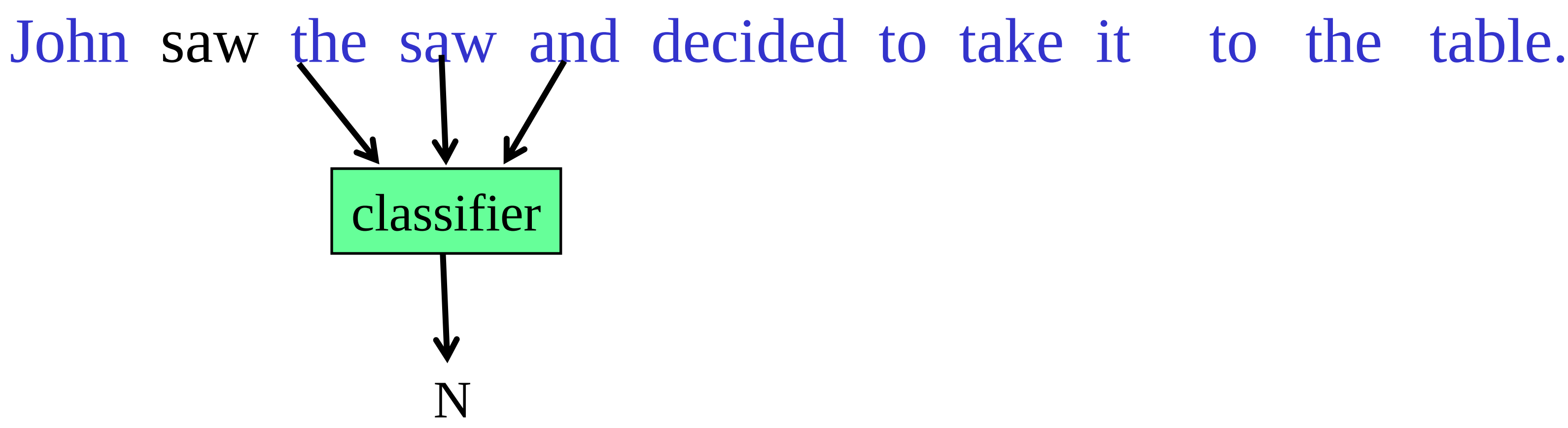
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification

Menti.com
7544 5229

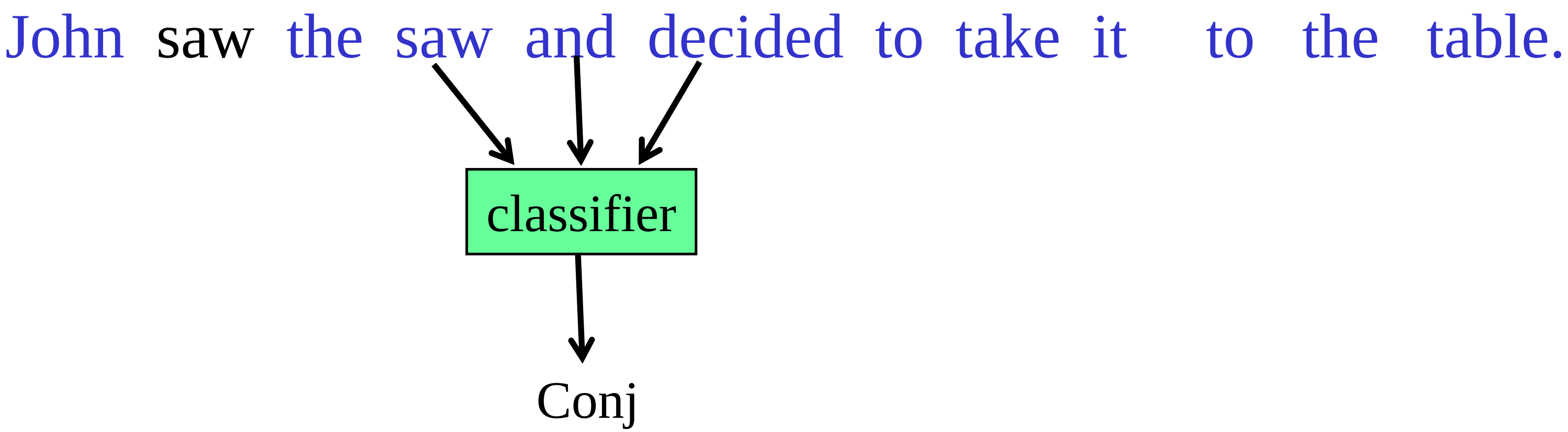
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification

Menti.com
7544 5229

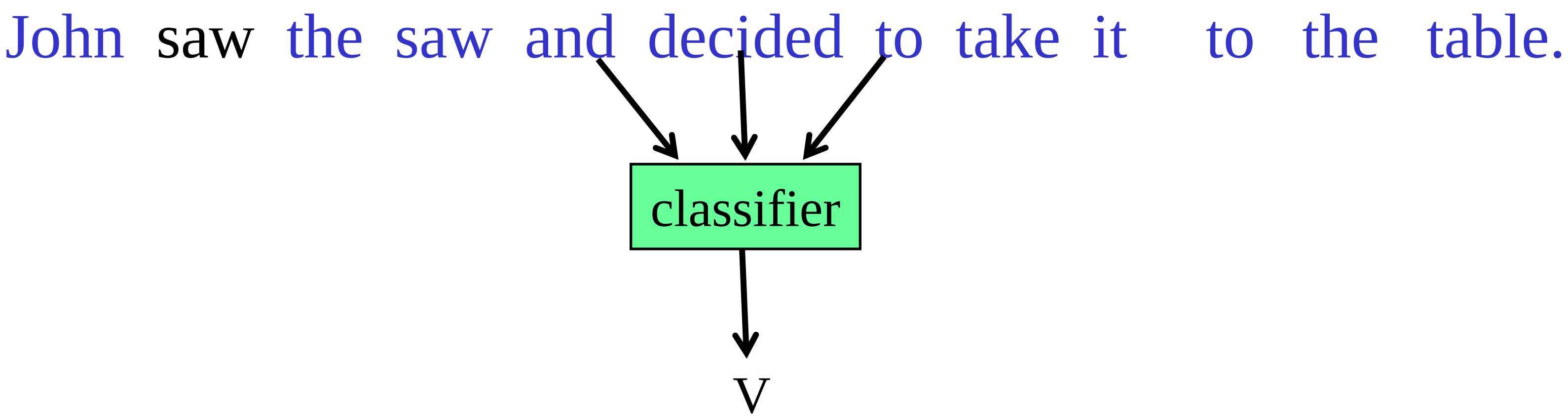
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



## Sequence Labeling as Classification

Menti.com
7544 5229

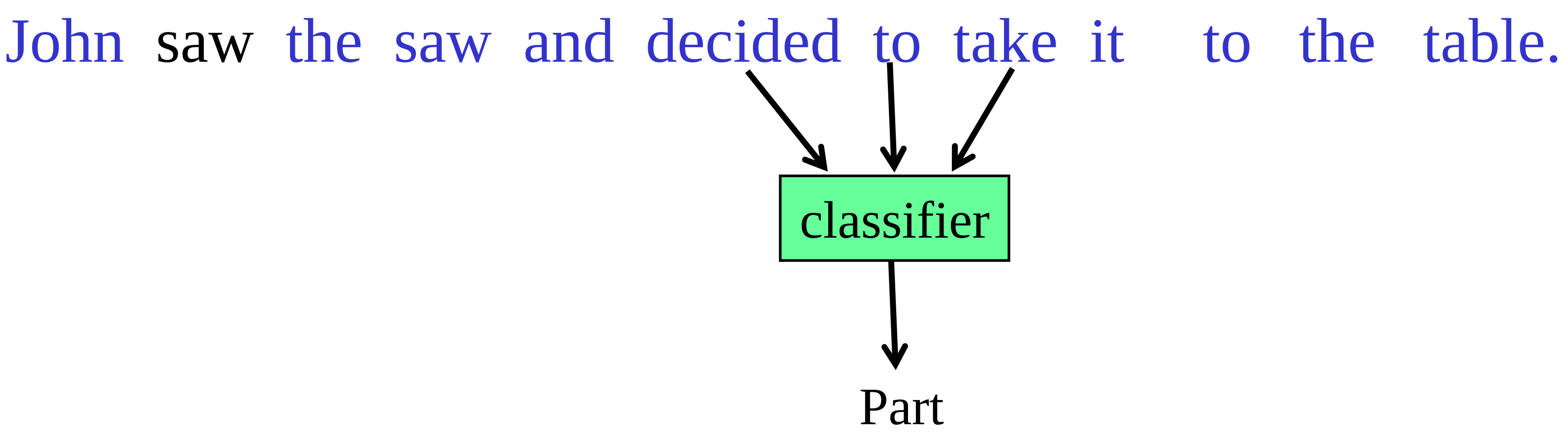
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



## Sequence Labeling as Classification

Menti.com
7544 5229

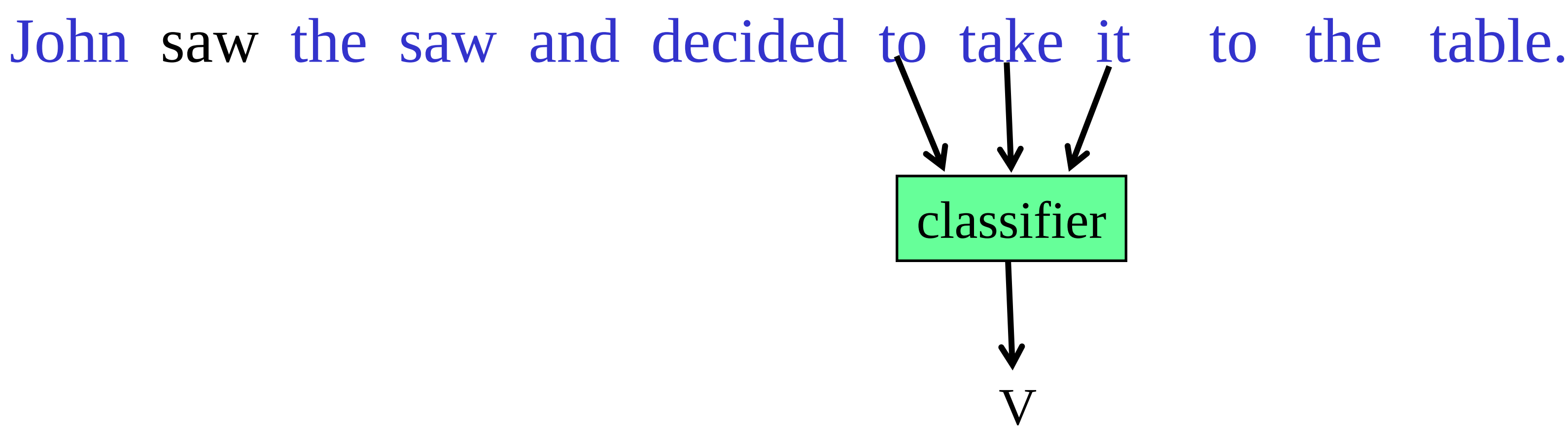
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification

Menti.com
7544 5229

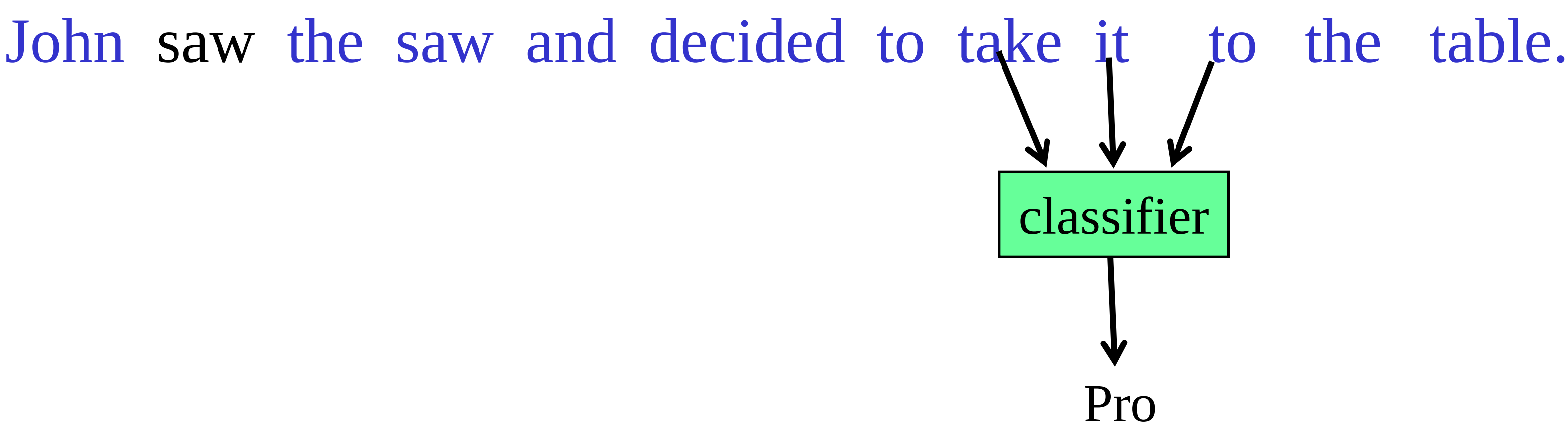
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Menti.com
7544 5229

# Sequence Labeling as Classification

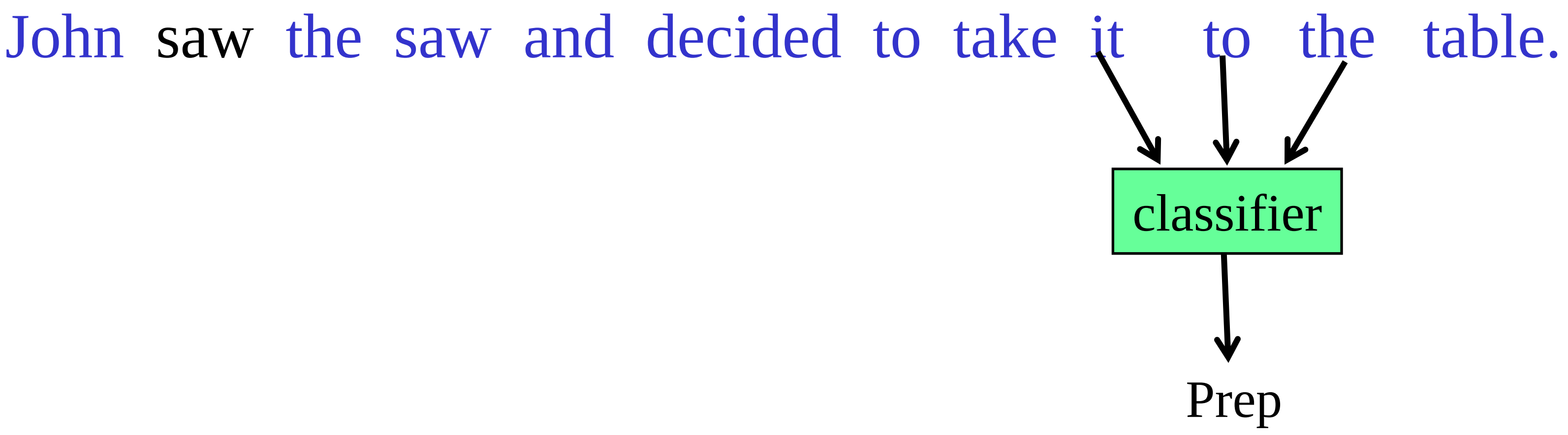
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification



- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

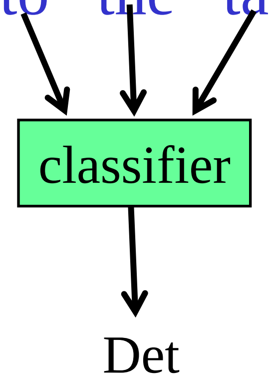


# Sequence Labeling as Classification

Menti.com
7544 5229

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

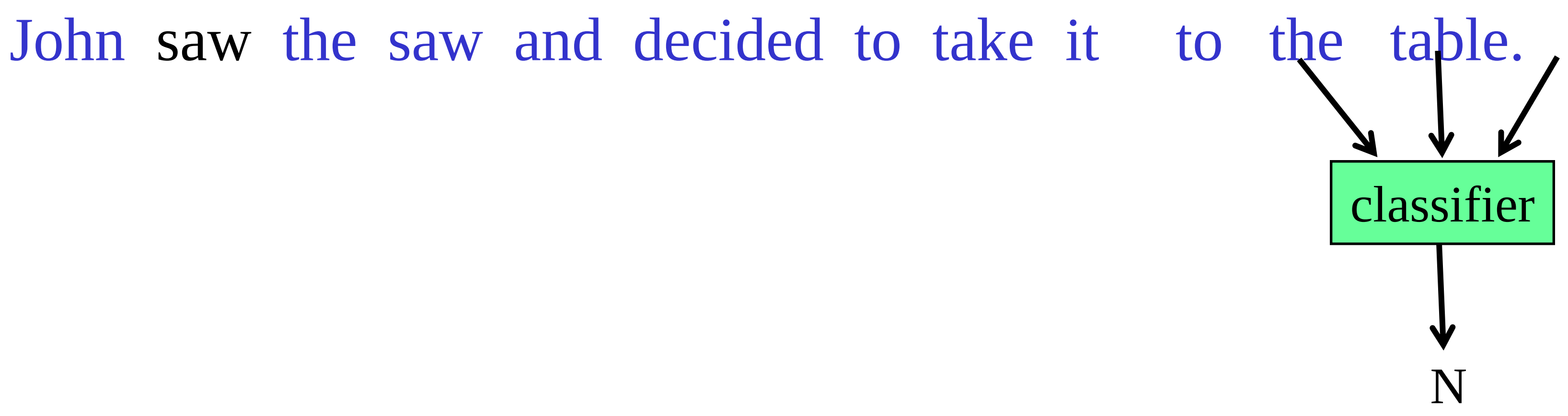
John saw the saw and decided to take it to the table.



## Sequence Labeling as Classification

Menti.com
7544 5229

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



# Sequence Labeling as Classification

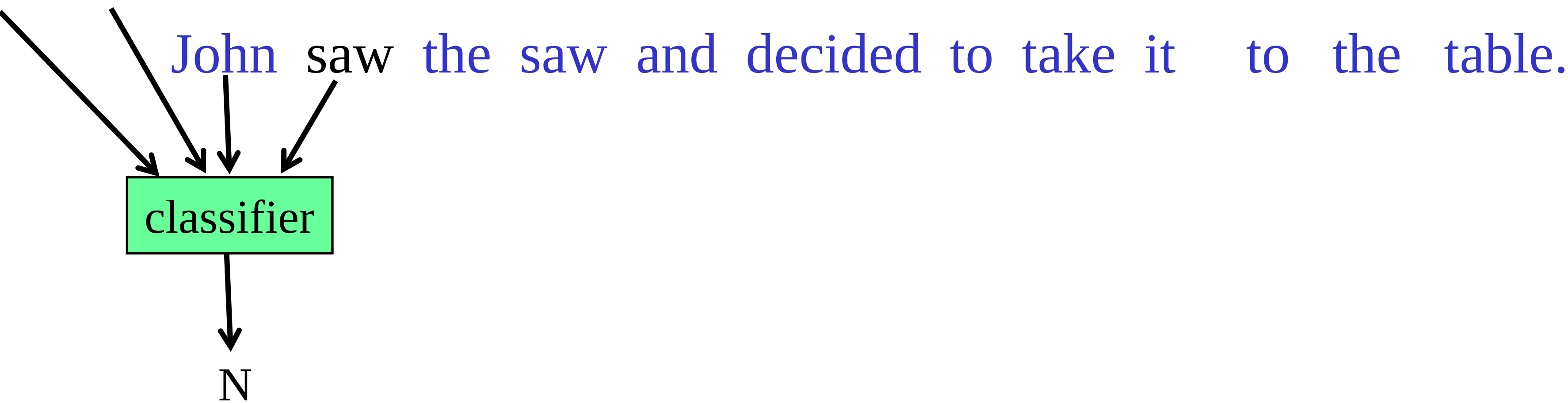
Menti.com
7544 5229

**Using Outputs as Inputs**

- Better input features are usually the categories of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

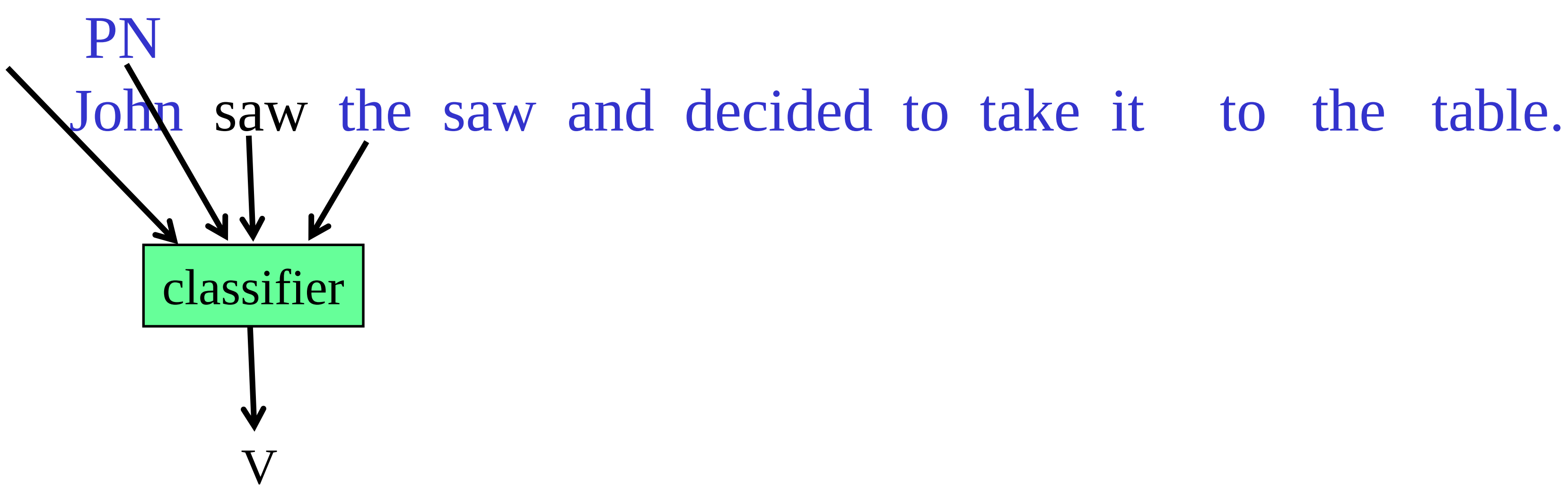
# Forward Classification

Menti.com
7544 5229



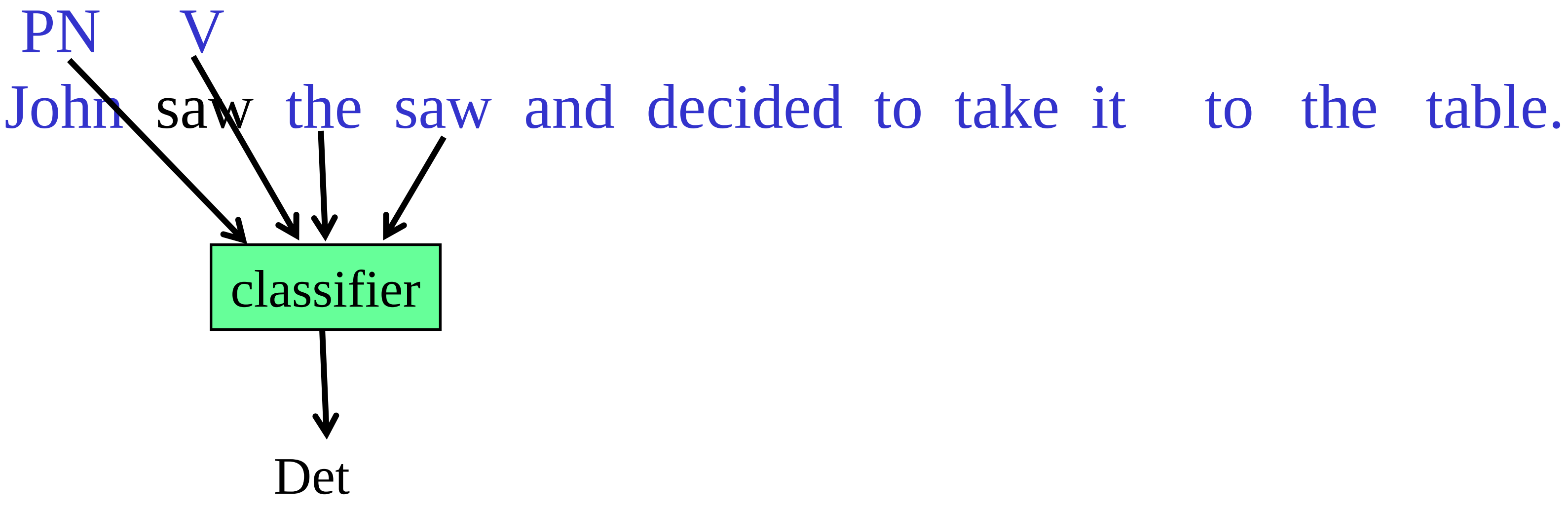
# Forward Classification

Menti.com
7544 5229



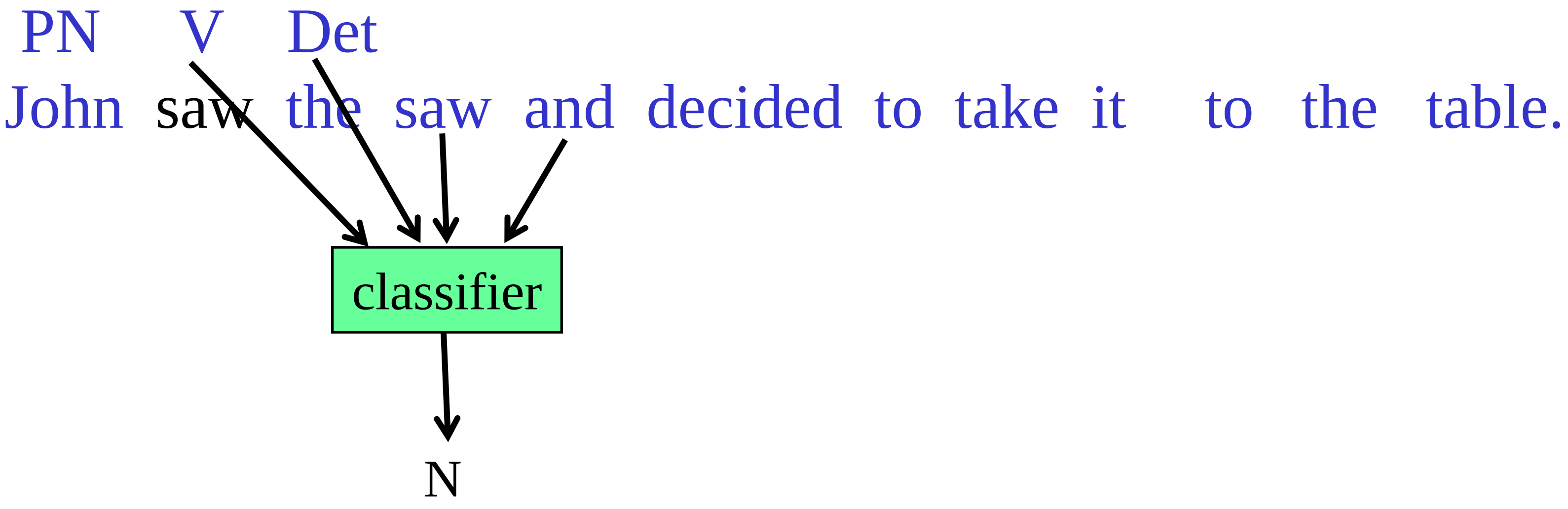
# Forward Classification

Menti.com
7544 5229



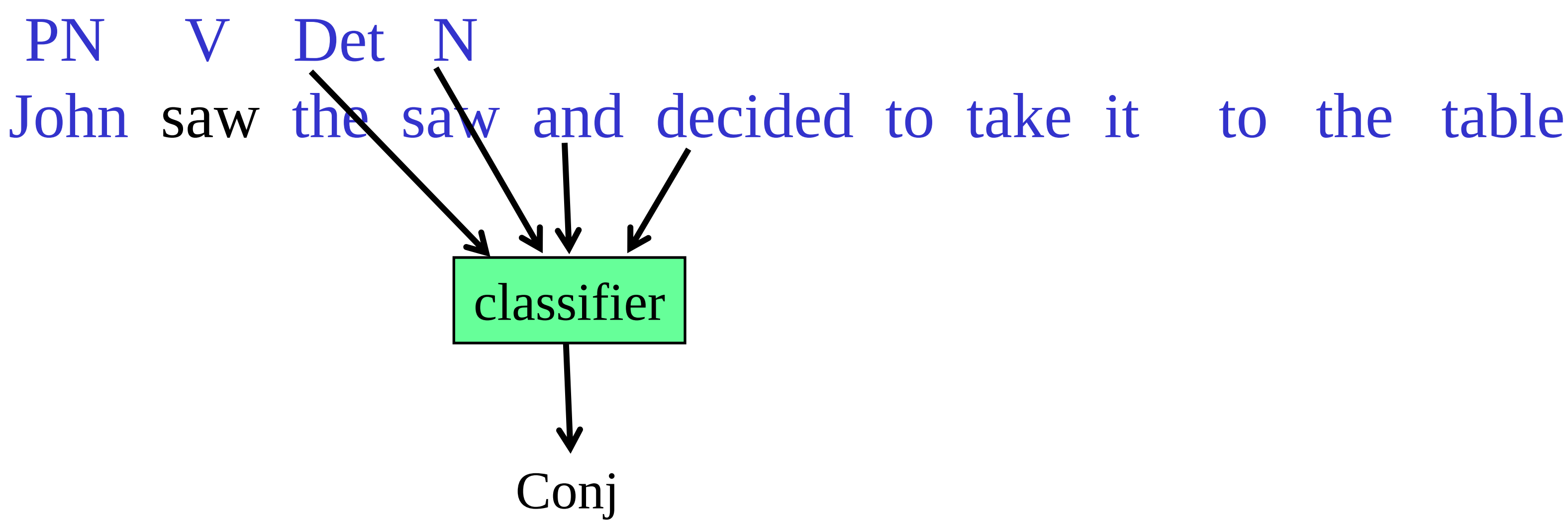
# Forward Classification

Menti.com
7544 5229



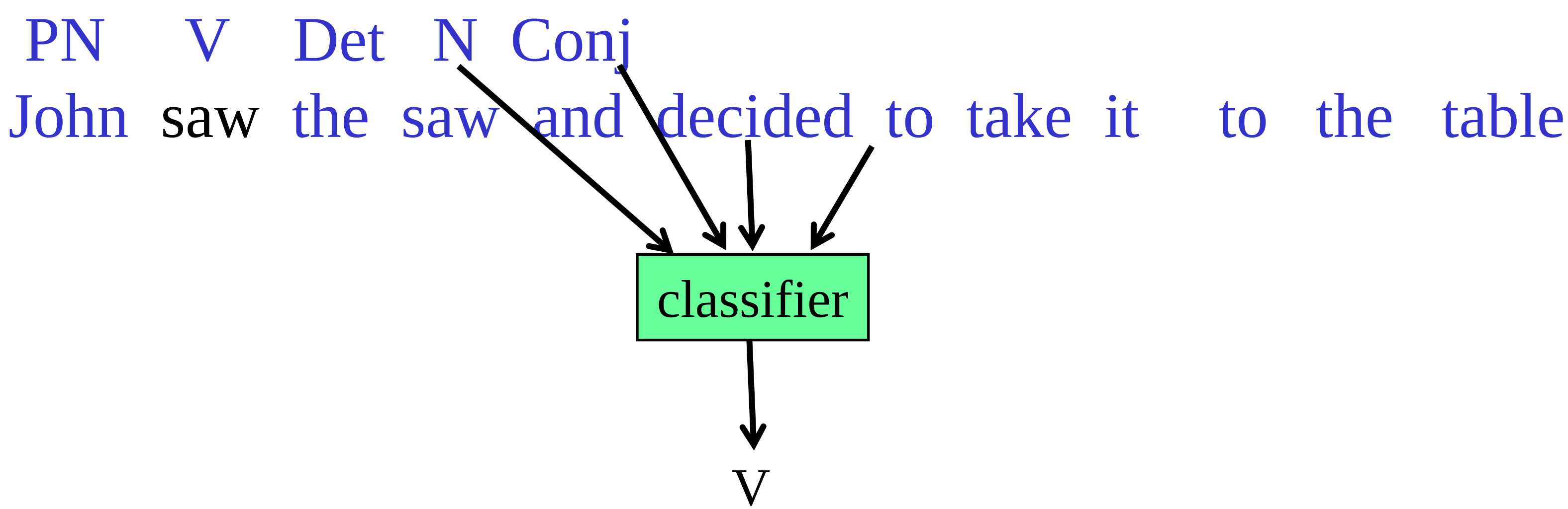
# Forward Classification

Menti.com
7544 5229



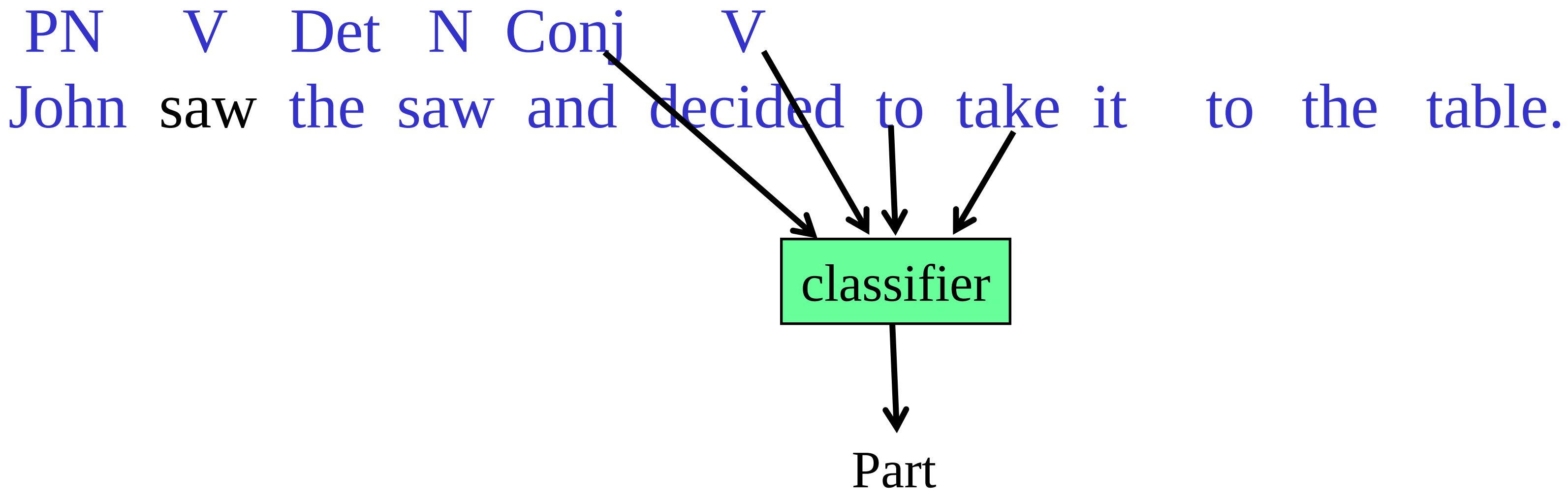
# Forward Classification

Menti.com
7544 5229



## Forward Classification





# Forward Classification

Menti.com
7544 5229

PN V Det N Conj V Part

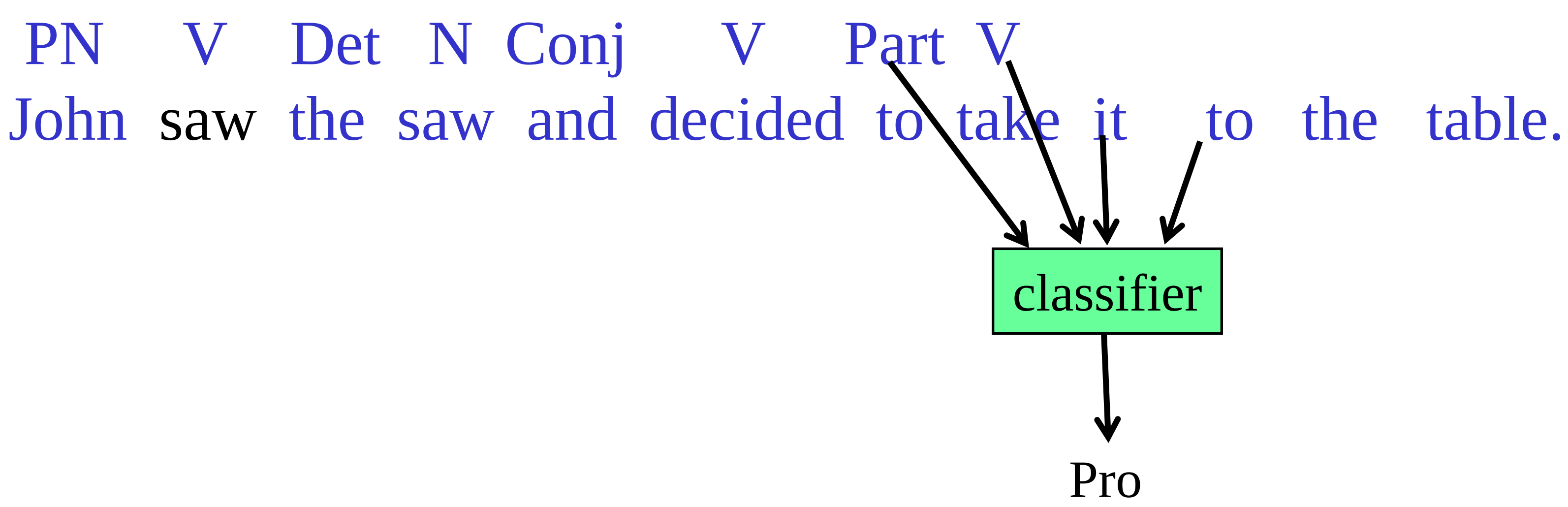
John saw the saw and decided to take it to the table.

classifier

V

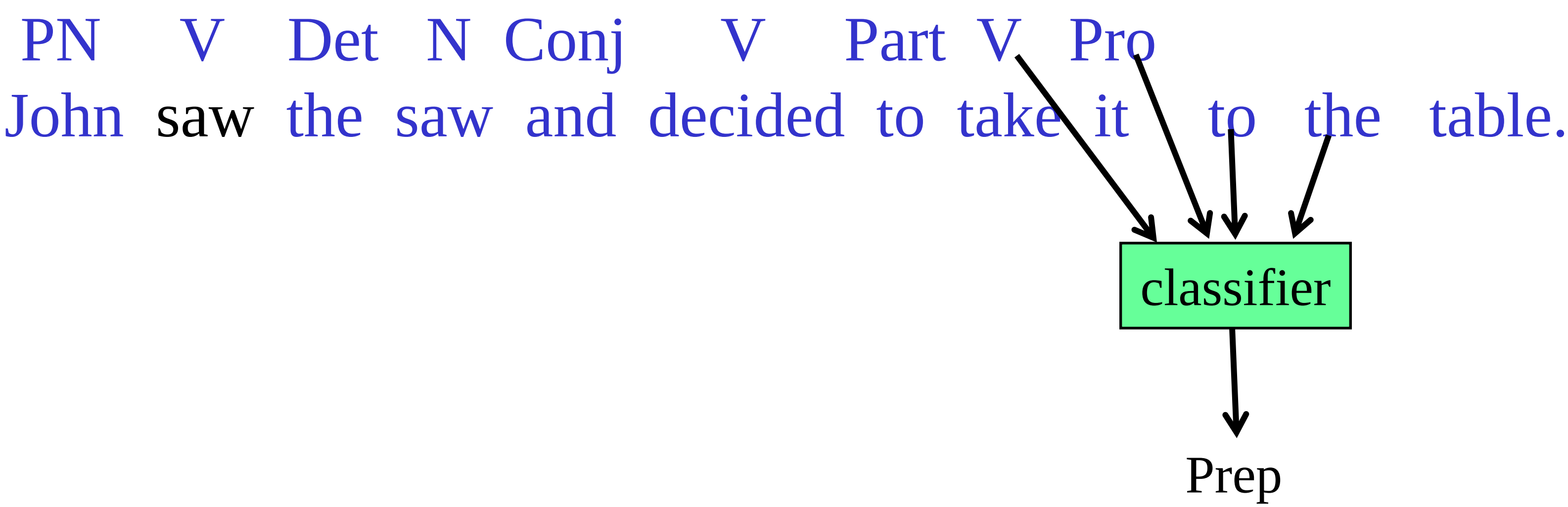
# Forward Classification

Menti.com
7544 5229



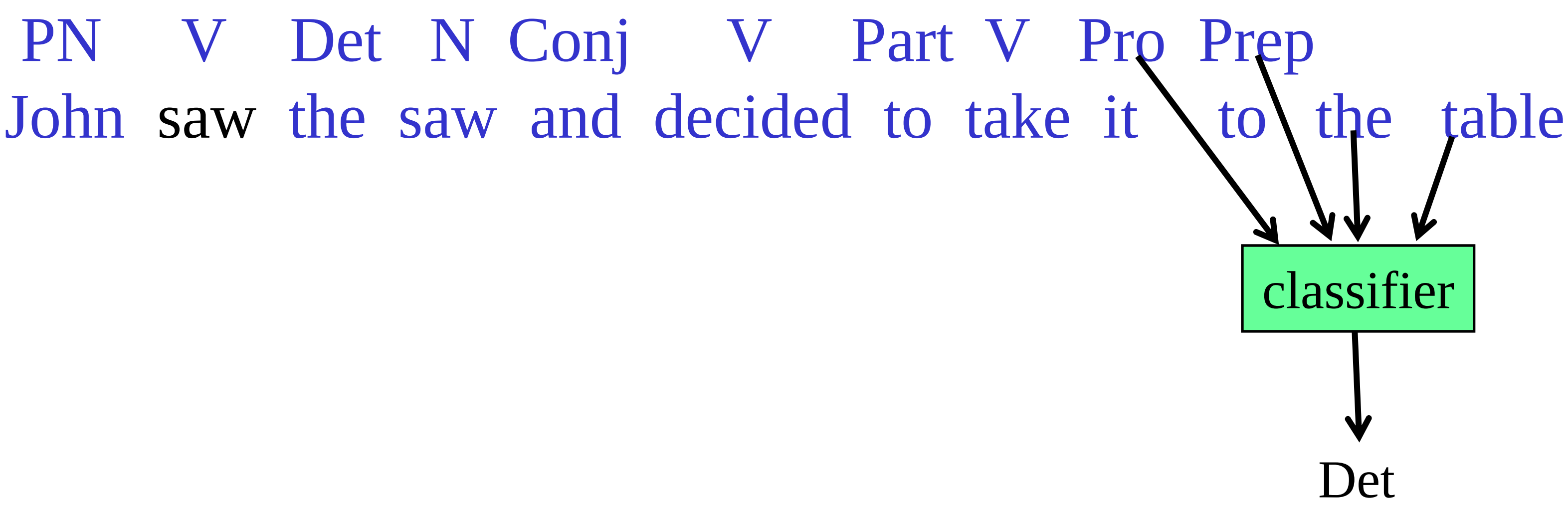
# Forward Classification

Menti.com
7544 5229



# Forward Classification



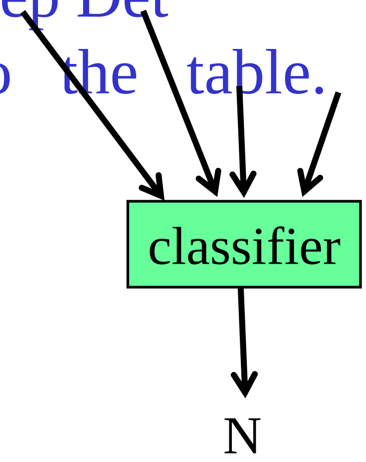


Menti.com
7544 5229

# Forward Classification

PN V Det N Conj V Part V Pro Prep Det

John saw the saw and decided to take it to the table.

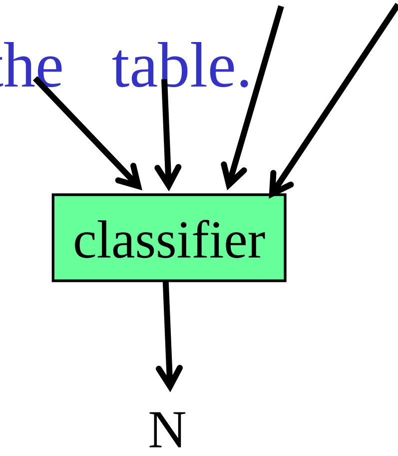


# Backward Classification

Menti.com
7544 5229

- Disambiguating “to” in this case would be even easier backward.

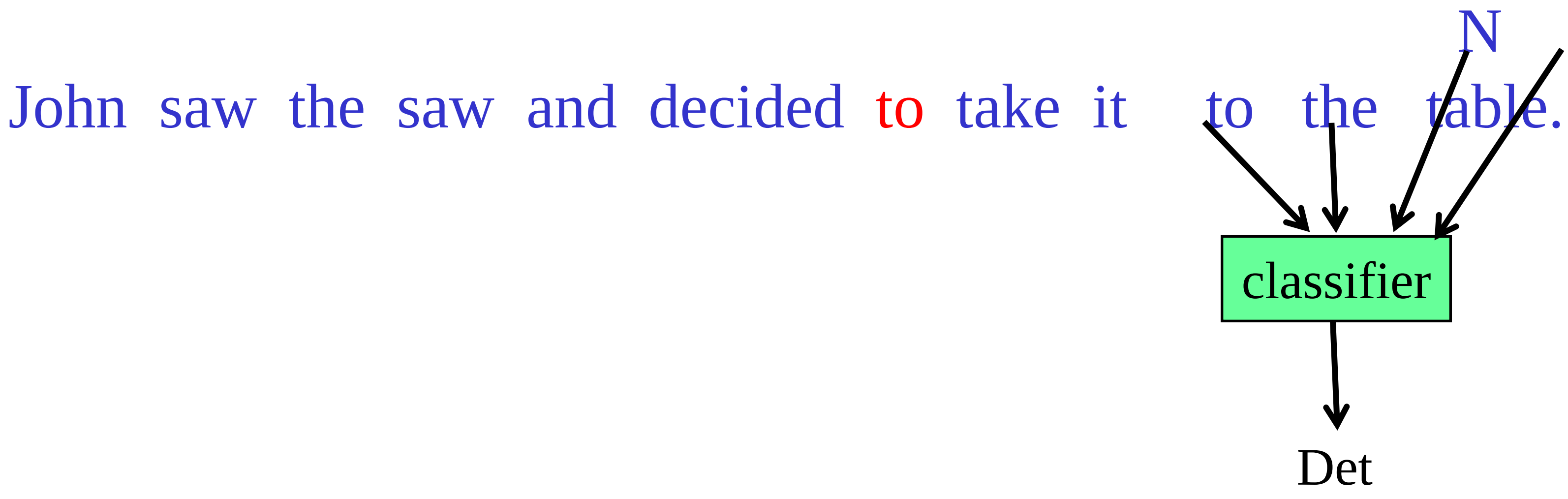
John saw the saw and decided to take it to the table.



# Backward Classification

Menti.com 7544 5229

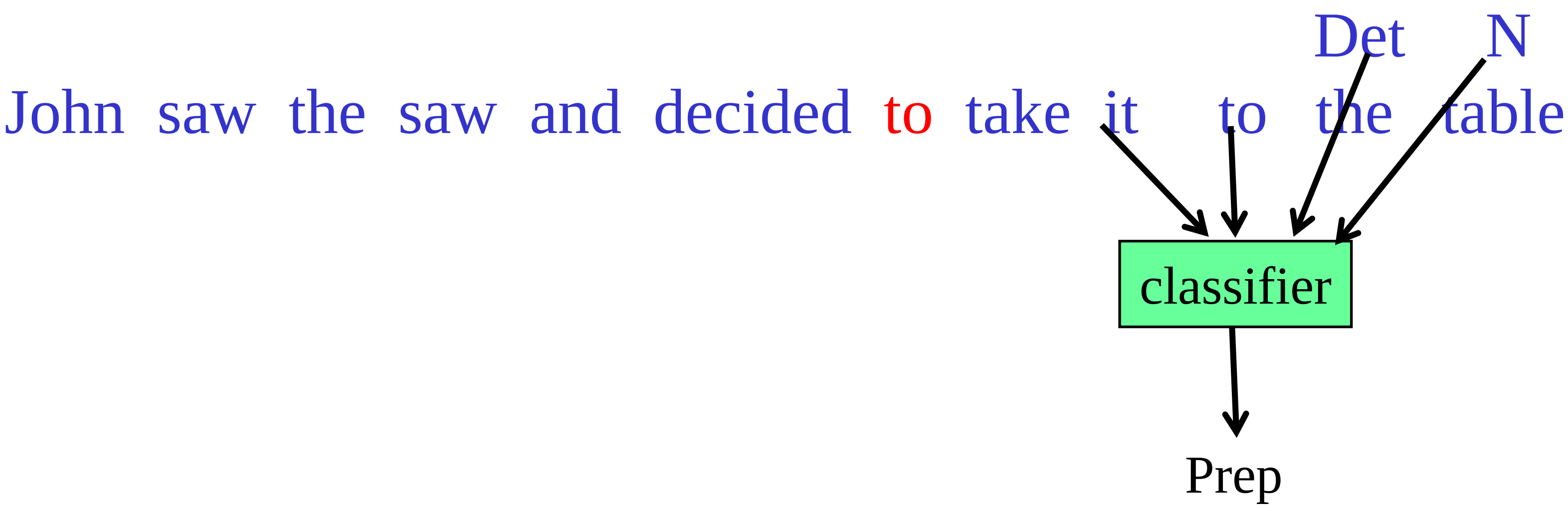
- Disambiguating “to” in this case would be even easier backward.



## Backward Classification

Menti.com
7544 5229

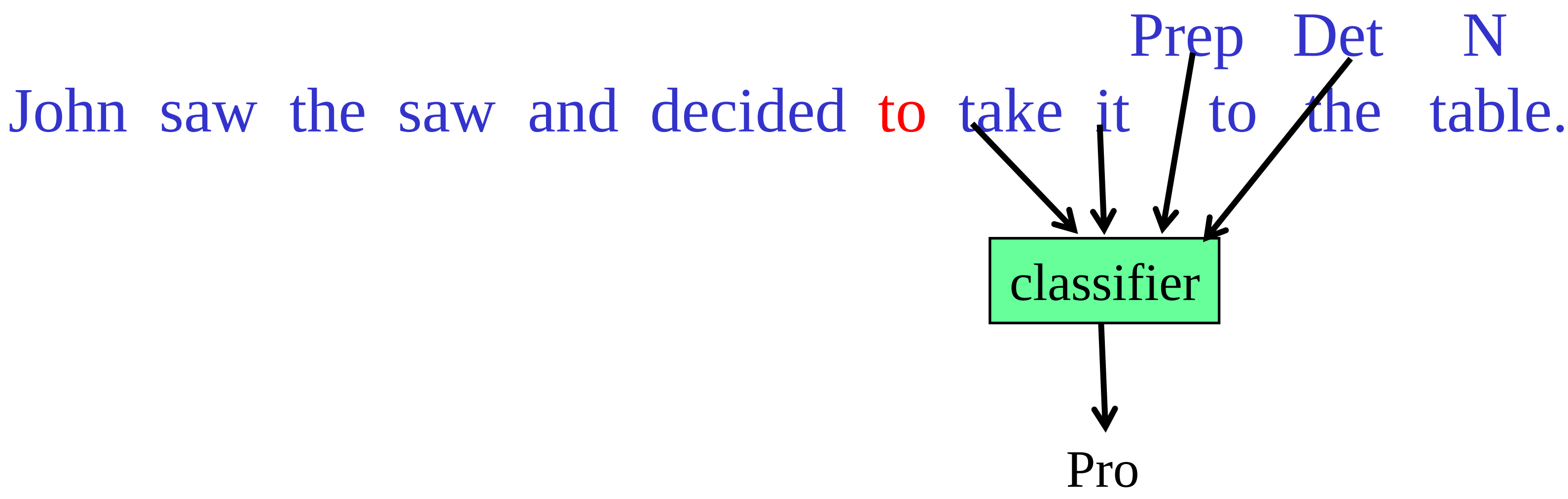
- Disambiguating “to” in this case would be even easier backward.



## Backward Classification

Menti.com
7544 5229

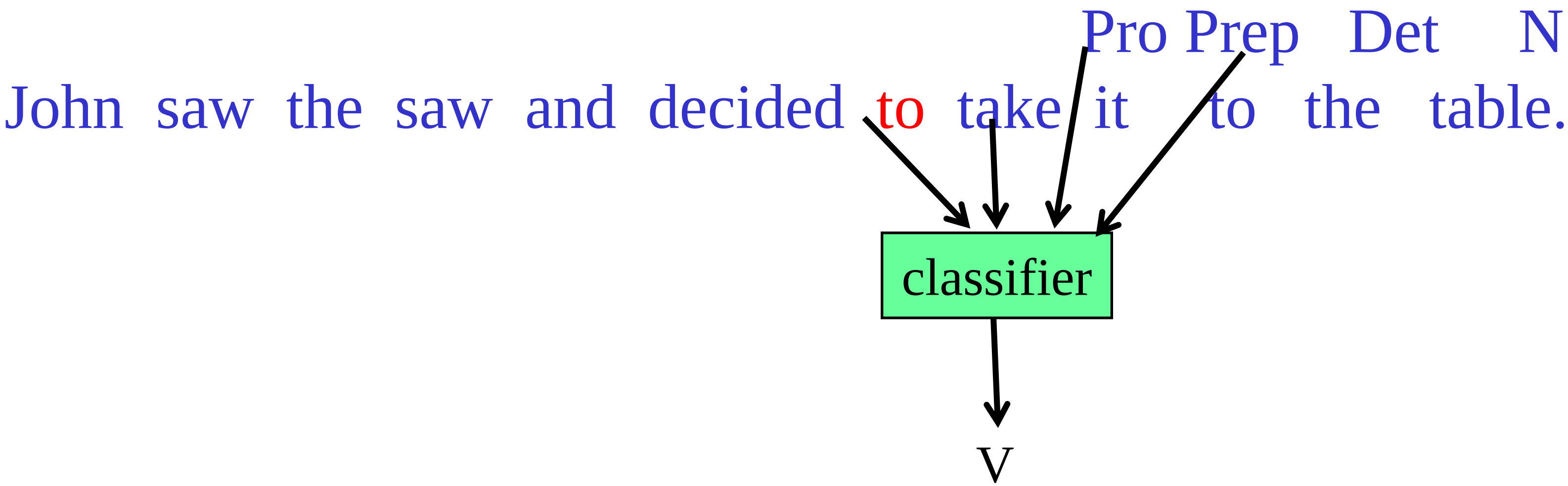
- Disambiguating “to” in this case would be even easier backward.



# Backward Classification

Menti.com
7544 5229

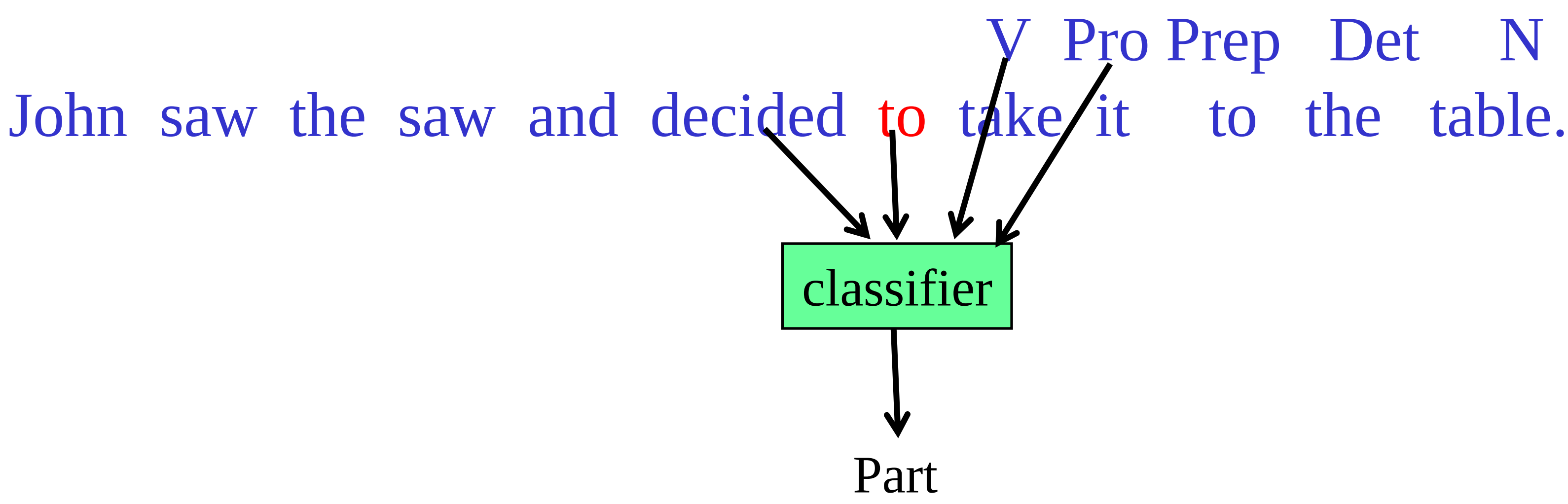
- Disambiguating “to” in this case would be even easier backward.



# Backward Classification

Menti.com
7544 5229

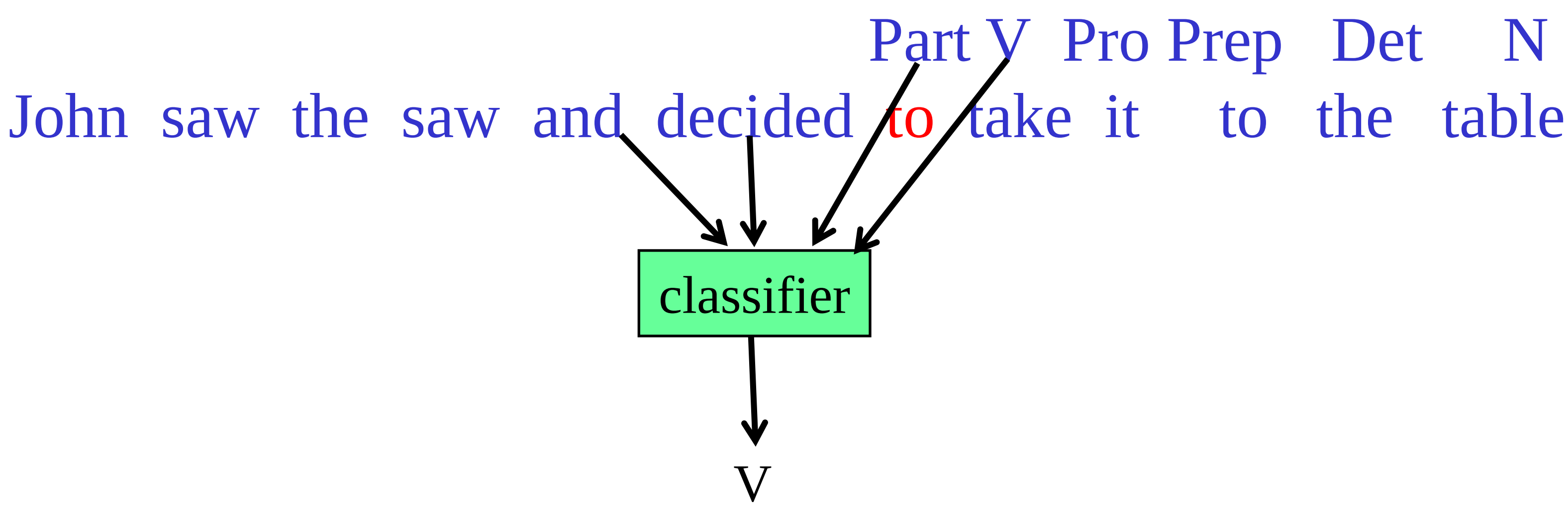
- Disambiguating “to” in this case would be even easier backward.
  - Preposition? (“I am heading to the store”)
  - Particle? (in front if infinitive verbs, like “I want to eat. I am going to leave.”)



## Backward Classification

Menti.com
7544 5229

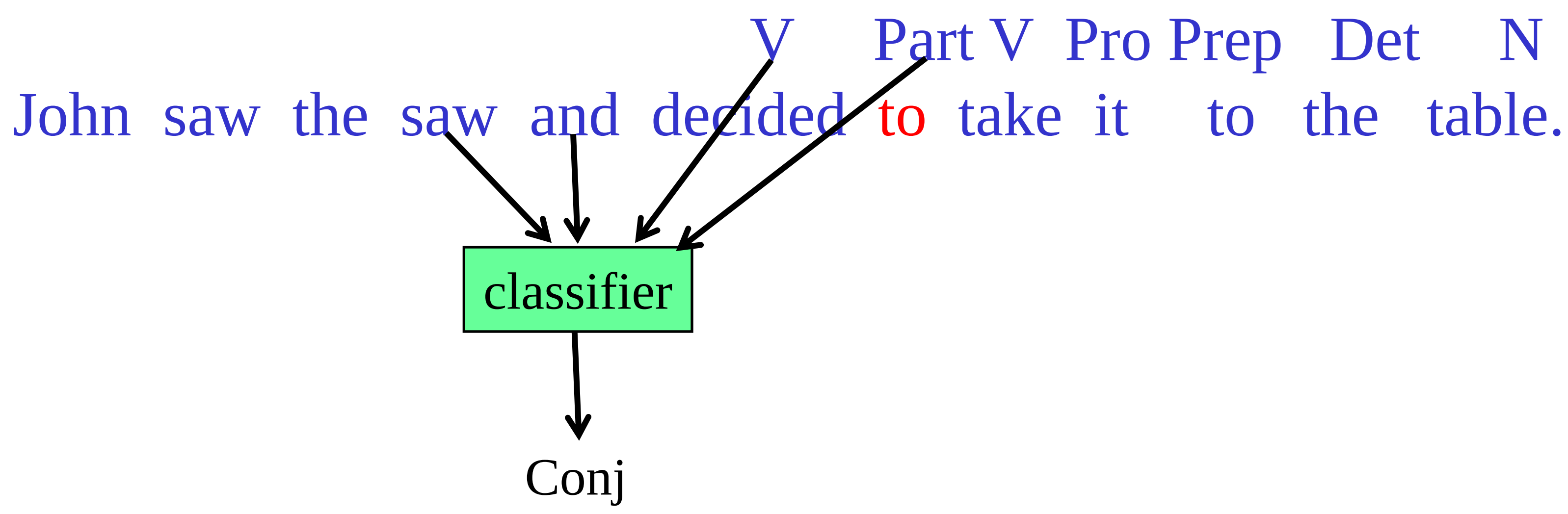
- Disambiguating “to” in this case would be even easier backward.



# Backward Classification



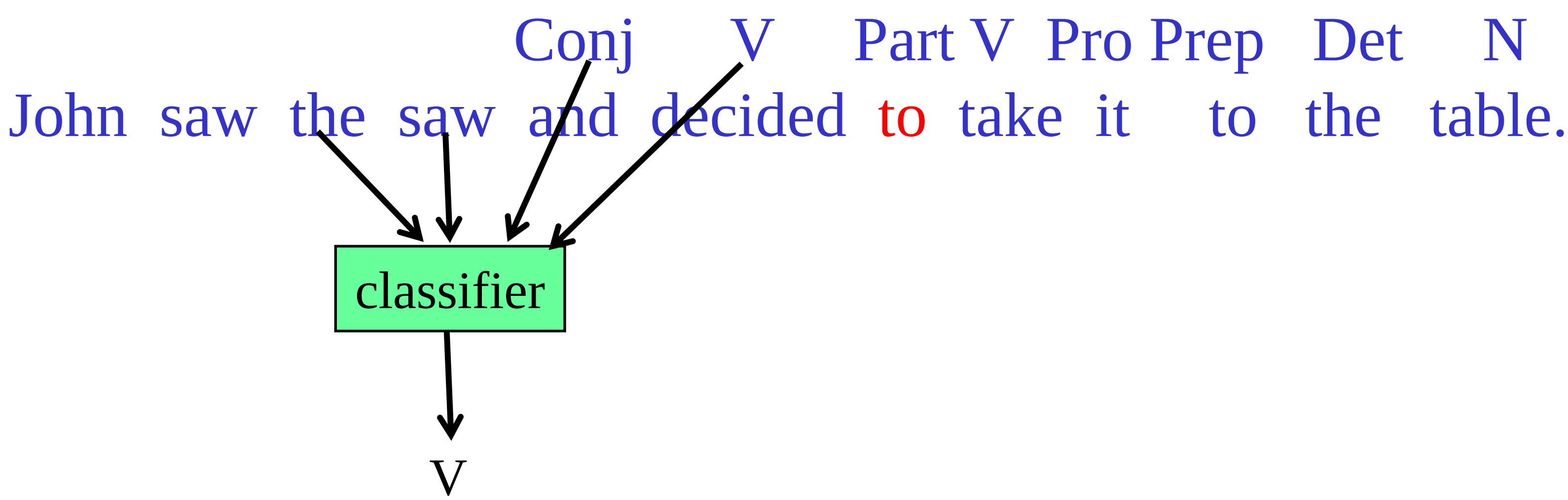
- Disambiguating “to” in this case would be even easier backward.



## Backward Classification

Menti.com
7544 5229

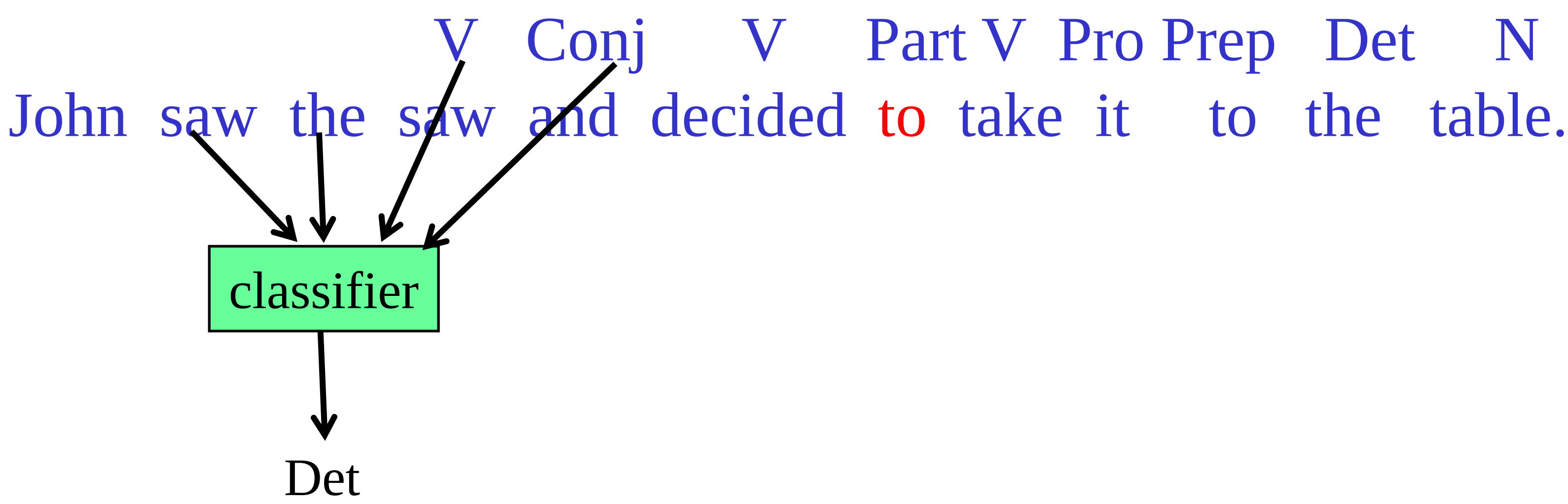
- Disambiguating “to” in this case would be even easier backward.



# Backward Classification

Menti.com
7544 5229

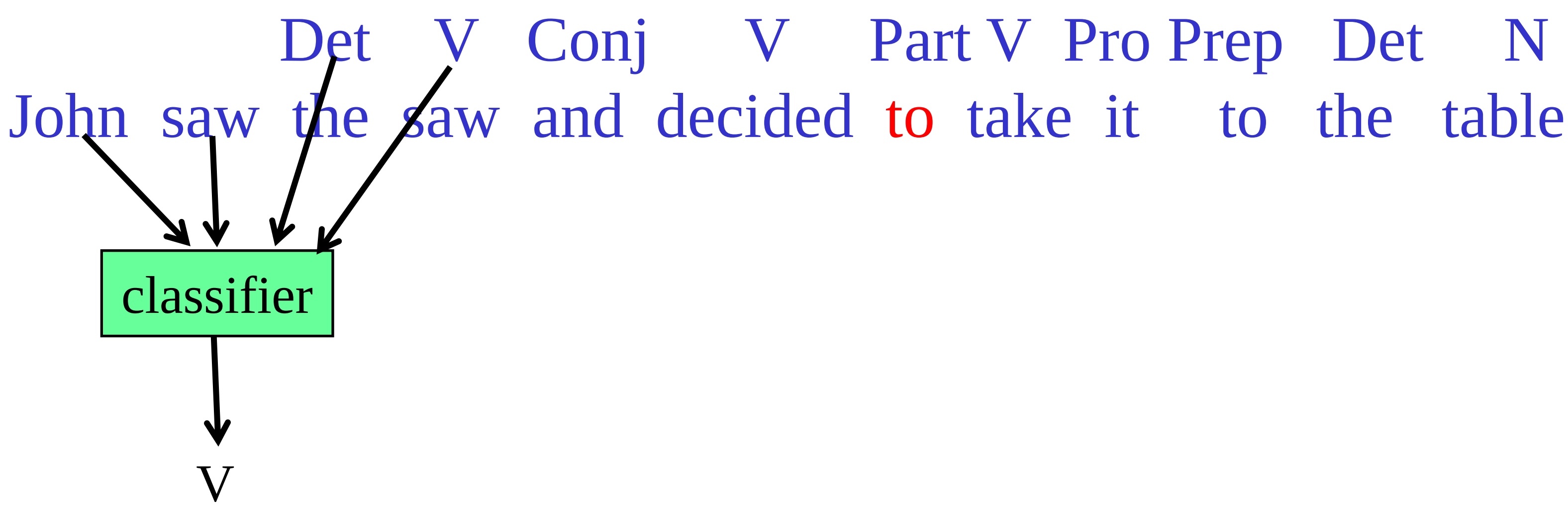
- Disambiguating “to” in this case would be even easier backward.



## Backward Classification

Menti.com
7544 5229

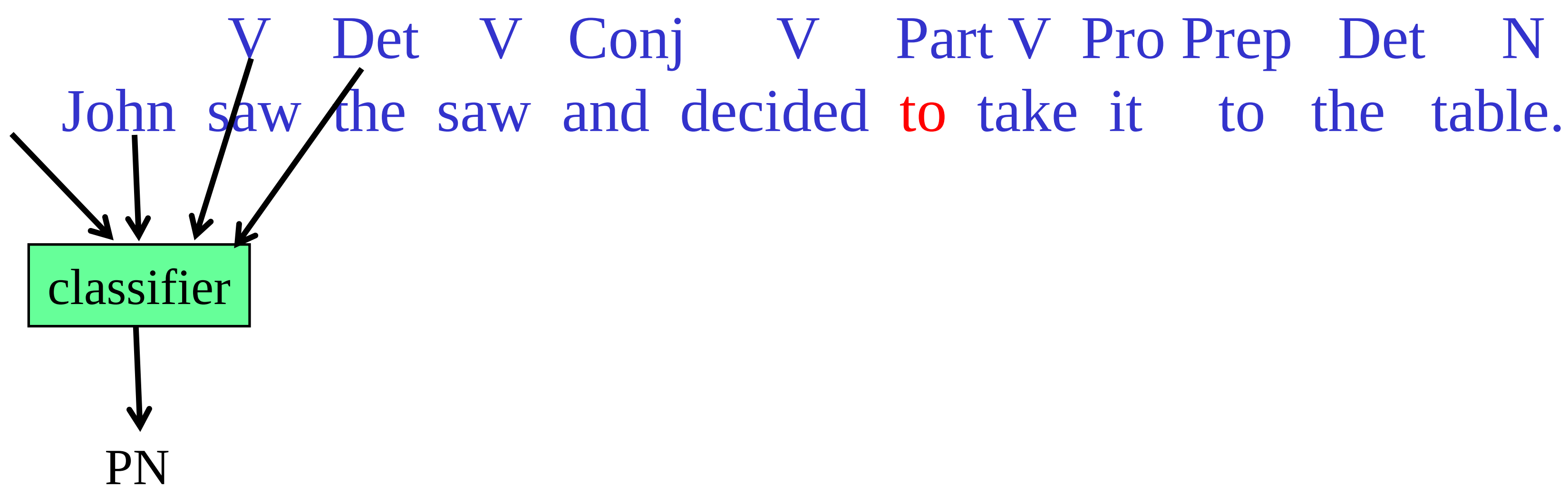
- Disambiguating “to” in this case would be even easier backward.



# Backward Classification

Menti.com
7544 5229

- Disambiguating “to” in this case would be even easier backward.



# Sequence Labeling as Classification

Menti.com
7544 5229

**Problems**

- Not easy to integrate information from category of tokens on both sides
- Difficult to propagate uncertainty between decisions and “collectively” determine the most likely joint assignment of categories to all of the tokens in a sequence.

# Probabilistic Sequence Models

Menti.com
7544 5229

- Probabilistic sequence models allow
  - integrating uncertainty over multiple, interdependent classifications
  - and collectively determine the most likely global assignment.

- Generative sequence models: e.g. **Hidden Markov Model (HMM)**

- Later in the lecture: Neural Sequence Models (RNN / LSTM)

# Intuition of HMM decoding: PoS Tagging

Menti.com
7544 5229

- Choose the tag sequence that is most probable given the observation sequence of n words:

Tag sequence

Best tag sequence $\boxed{\hat{t}_1^n} = \underset{t_1^n}{\operatorname{argmax}} P(\boxed{t_1^n}|\boxed{w_1^n})$ Word sequence

- Bayes' Rule:,

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

- Drop the denominator

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

- Assumption 1: word appearing depends only on its own tag

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

- Assumption 2: the probability of a tag is dependent only on the previous tag

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

# Hidden Markov Model

Menti.com
7544 5229

A Hidden Markov Model is a statistical model of hidden, stochastic state transitions with observable, stochastic output. Key features:

- A fixed set of states
  - At each "time", the hidden markov model is in exactly one of these states
- State <u>transition probabilities</u>
  - The starting state can be fixed or probabilistic
- A fixed set of possible outputs
- For each state: a distribution of probabilities for every possible output
  - Also called <u>emission probabilities</u>

Task:

For an observed output sequence, what is the (hidden) state sequence that has the highest probabiliy to produce this output?

# Hidden Markov Model - Example



Every day, Darth Vader is in one of three moods: Good, Neutral or Bad

But, because he wears his mask, we cannot observe it!

# Hidden Markov Model - Example

Menti.com
7544 5229

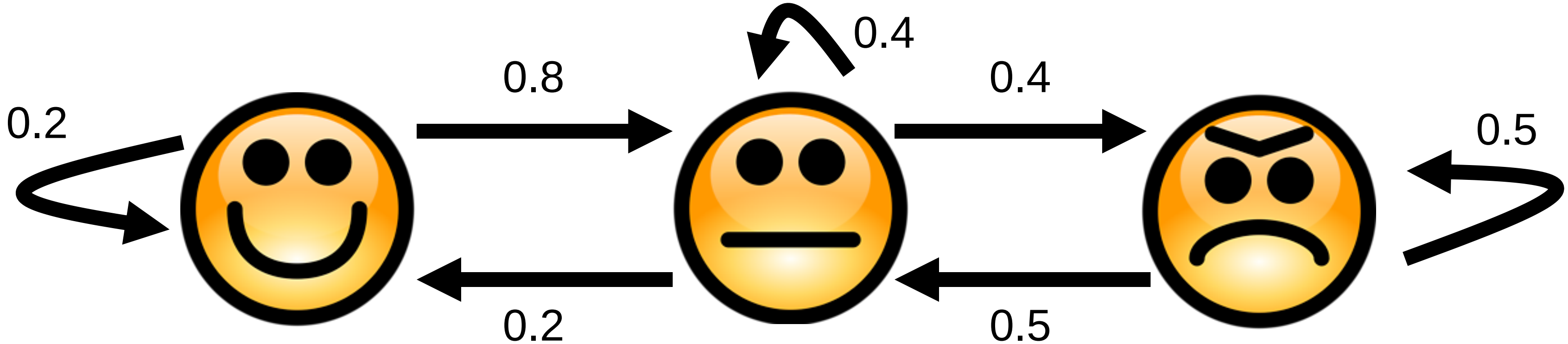
Every day, Darth Vader is in one of three moods: Good, Neutral or Bad

But, because he wears his mask, we cannot observe it!



# Hidden Markov Model - Example

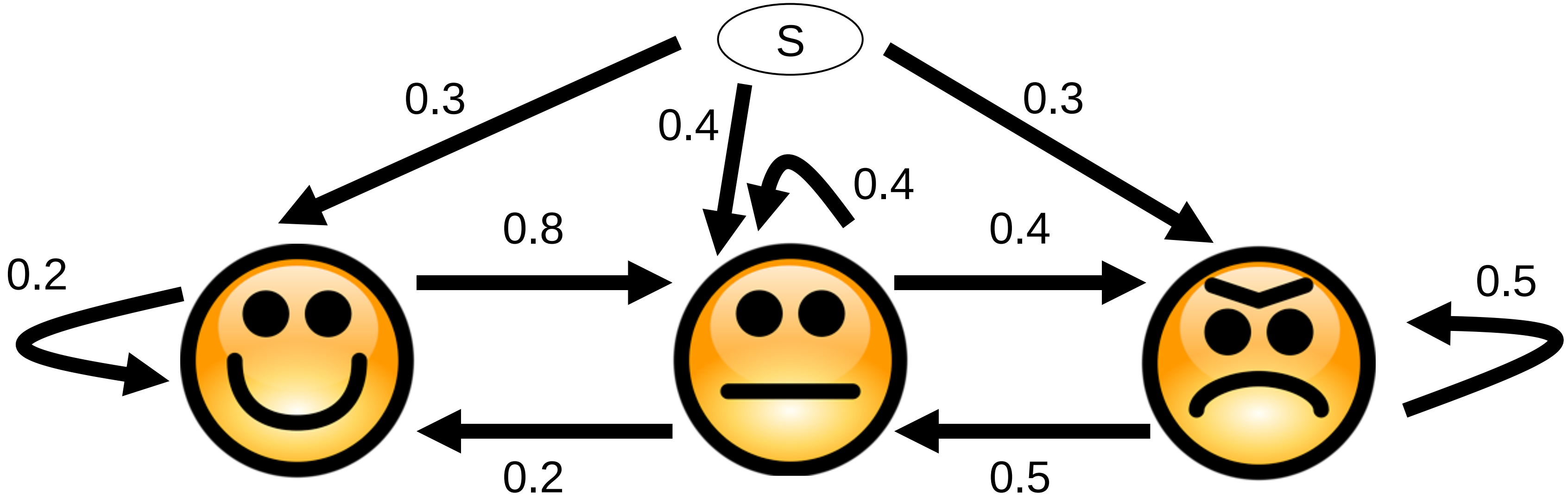

Somehow, you know the odds how his mood changes from day to day:



# Hidden Markov Model - Example

Menti.com
7544 5229

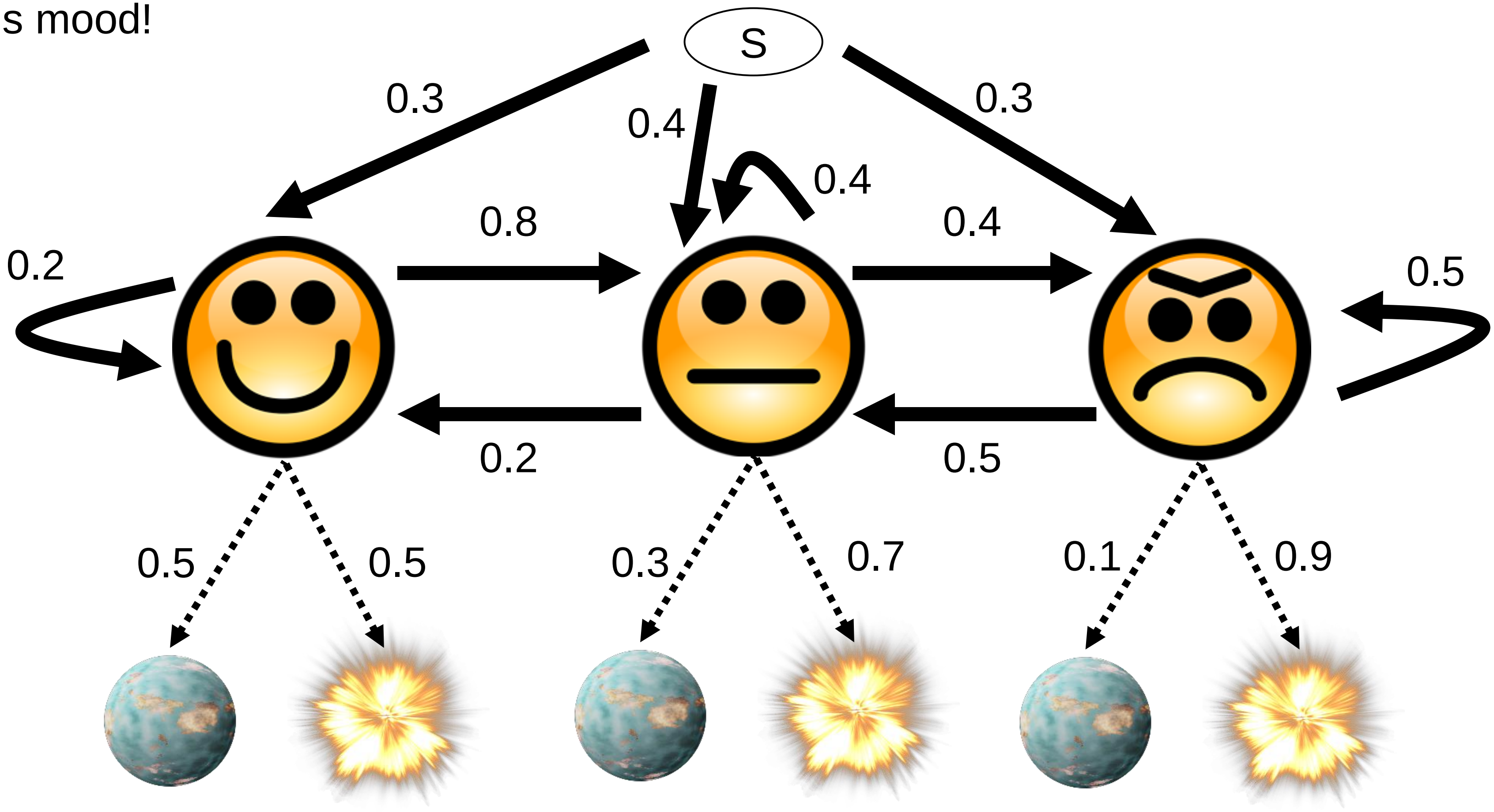
You also know the chances for his mood on day 1:



# Hidden Markov Model - Example

Menti.com
7544 5229

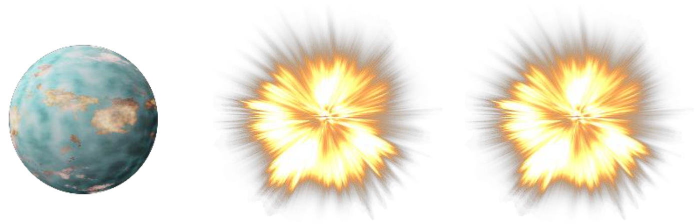
What we CAN observe is if Darth Vader destroys a planet or not, which depends on his mood!



# Hidden Markov Model - Example

Menti.com
7544 5229

We observe that he does not destroy a planet on the first day, but he destroys a planet each on the second and third day:



Question:

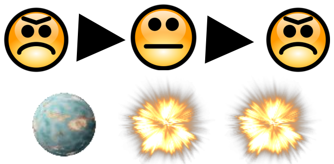
What is the most probable sequence of his mood on these three days?

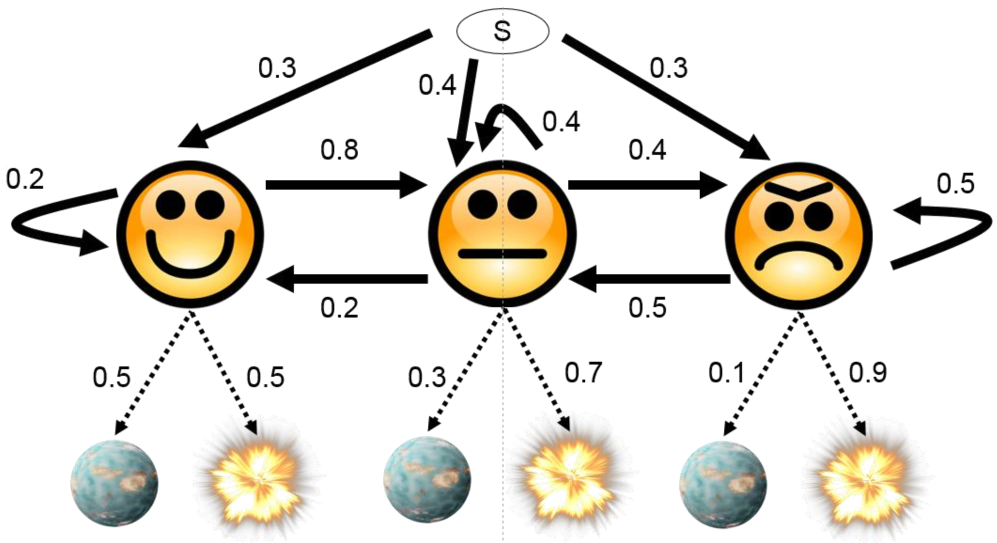
?

# Hidden Markov Model - Example

Menti.com
7544 5229

What is the probability for this mood sequence and this observation:

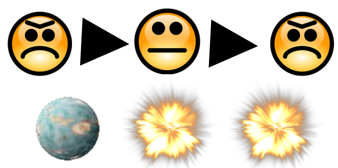


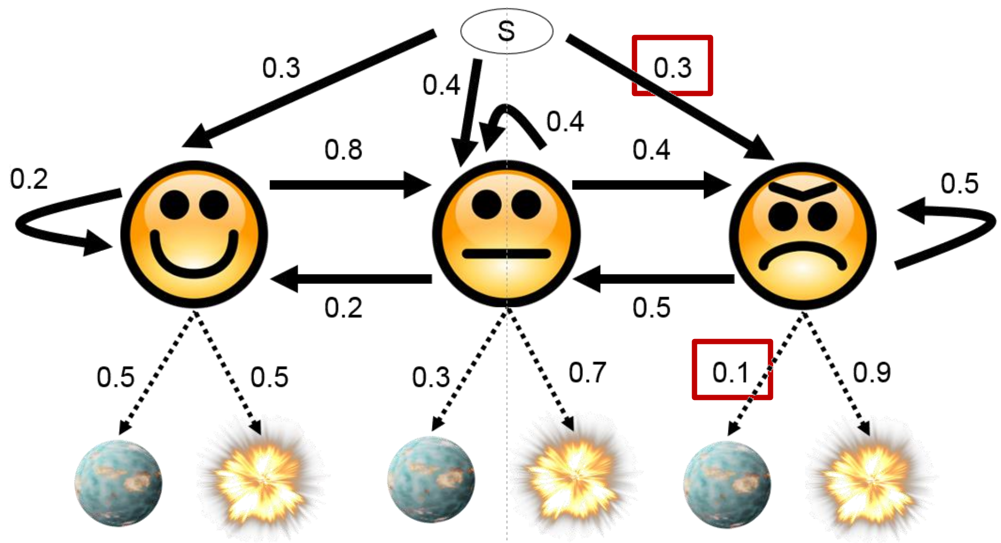


# Hidden Markov Model - Example

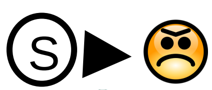
Menti.com
7544 5229

What is the probability for this mood sequence and this observation:






First day:

Transition probability:  0.3

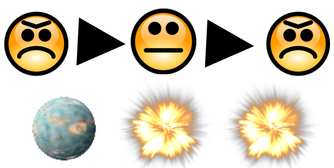
Emission probability:  0.1

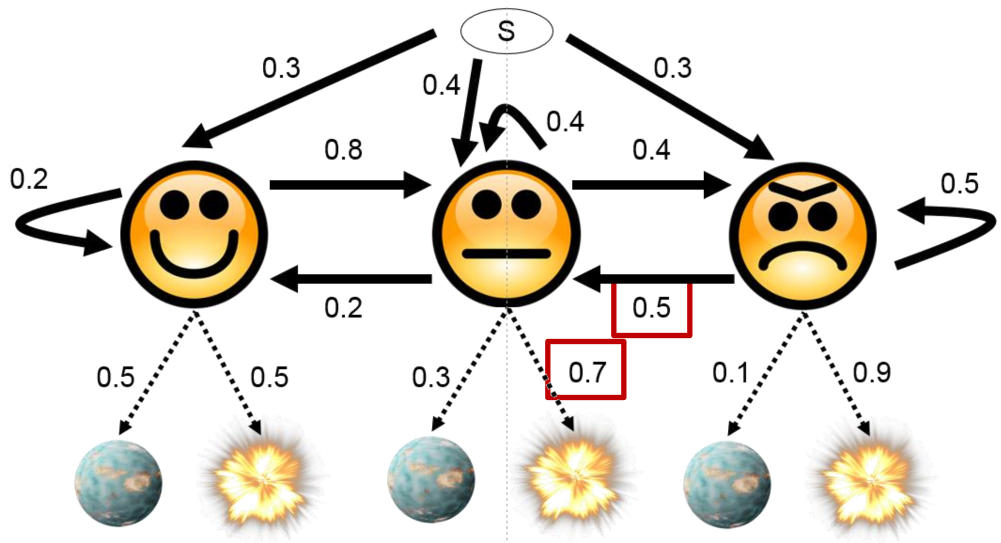
Joint probability: 0.3 * 0.1 = 0.03

# Hidden Markov Model - Example

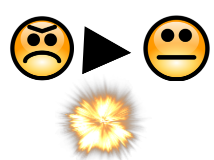
Menti.com
7544 5229

What is the probability for this mood sequence and this observation:



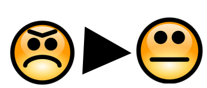



Second day:

Transition probability:  0.5

Emission probability: 0.7

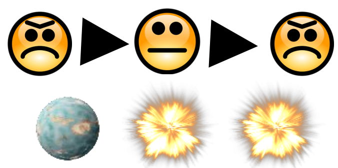
Joint probability: 0.5 * 0.7 = 0.35

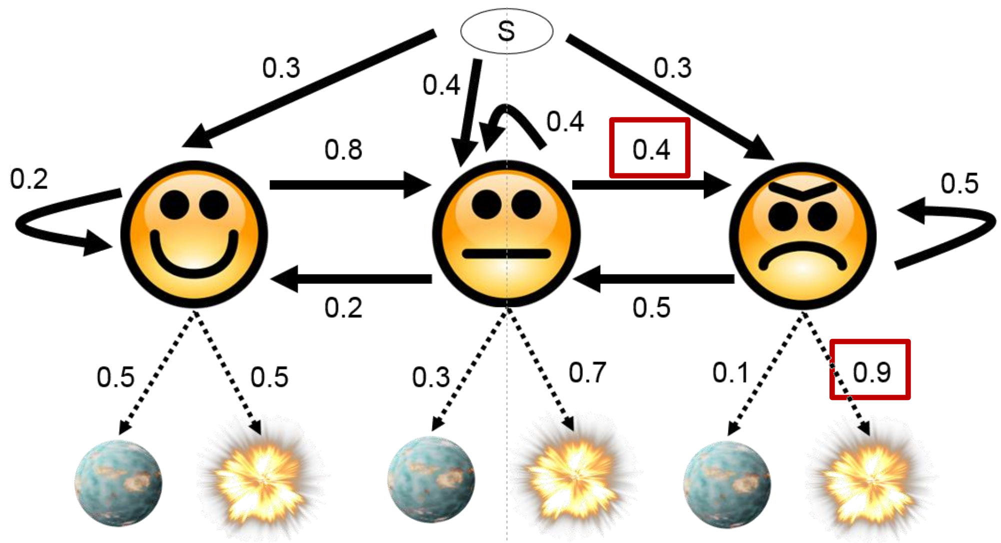
Probability for sequence:  0.03 (day one) * 0.35 (day two) = 0.0105

# Hidden Markov Model - Example

Menti.com
7544 5229

What is the probability for this mood sequence and this observation:



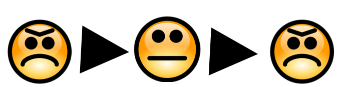



Third day:

Transition probability:  0.4

Emission probability:  0.9

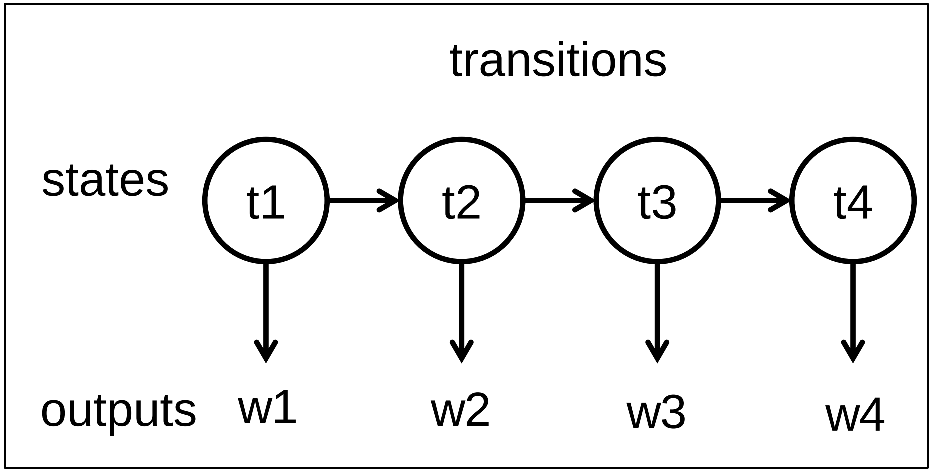
Joint probability: 0.4 * 0.9 = 0.36

Probability for sequence:  0.03 * 0.35 * 0.36 = 0.00378

# Hidden Markov Model – Application to NLP

Menti.com
7544 5229

In our POS tagging example, we know the sequence of words, and we want to know the sequence of POS tags!

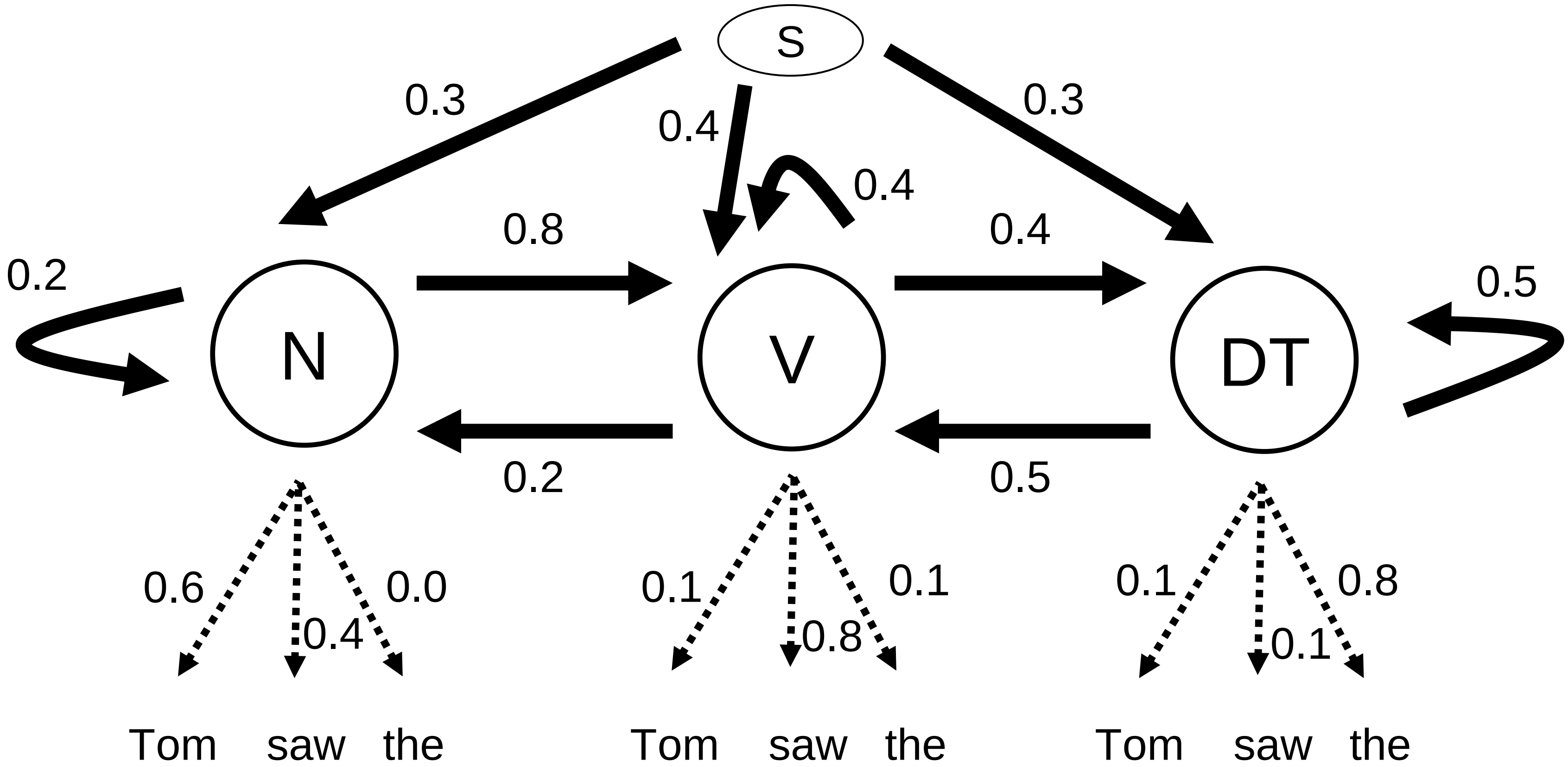



- (hidden) States: POS tags
- (observable) Outputs: Tokens
- We also need
  - Transition probabilities between states
    - Also: Initial probabilities = Probabilities for the first token of a sentence
  - Emission probabilities for every state

How would our graph from the example look like?

# Hidden Markov Model – Application to NLP

Menti.com
7544 5229




How do we get to the transition and emission probabilities?

# Hidden Markov Model – Application to NLP: Estimating the Probabilities

Menti.com
7544 5229

- The probabilities are estimated just by counting on a **tagged training corpus**
  - Transition probability: how often the first tag is followed by the second divided by the number of the times the first tag was seen in a labeled corpus

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

  - The emission probabilities: the number of times the word was associated with the tag in the labeled corpus divided by number of the times the first tag was seen in a labeled corpus

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

# Example

Menti.com
7544 5229

Input sentence: Janet will back the bill
Correct PoS tags: Janet/NNP will/MD back/VB the/DT bill/NN

Transition probabilities based on WSJ corpus
Rows are labeled with the conditioning event; thus P(VB|MD) is 0.7968.

| | NNP | MD | VB | JJ | NN | RB | DT |
|---|---|---|---|---|---|---|---|
| $<s>$ | 0.2767 | 0.0006 | 0.0031 | 0.0453 | 0.0449 | 0.0510 | 0.2026 |
| **NNP** | 0.3777 | 0.0110 | 0.0009 | 0.0084 | 0.0584 | 0.0090 | 0.0025 |
| **MD** | 0.0008 | 0.0002 | 0.7968 | 0.0005 | 0.0008 | 0.1698 | 0.0041 |
| **VB** | 0.0322 | 0.0005 | 0.0050 | 0.0837 | 0.0615 | 0.0514 | 0.2231 |
| **JJ** | 0.0366 | 0.0004 | 0.0001 | 0.0733 | 0.4509 | 0.0036 | 0.0036 |
| **NN** | 0.0096 | 0.0176 | 0.0014 | 0.0086 | 0.1216 | 0.0177 | 0.0068 |
| **RB** | 0.0068 | 0.0102 | 0.1011 | 0.1012 | 0.0120 | 0.0728 | 0.0479 |
| **DT** | 0.1147 | 0.0021 | 0.0002 | 0.2157 | 0.4744 | 0.0102 | 0.0017 |

Given the observation (output) likelihoods

| | Janet | will | back | the | bill |
|---|---|---|---|---|---|
| **NNP** | 0.000032 | 0 | 0 | 0.000048 | 0 |
| **MD** | 0 | 0.308431 | 0 | 0 | 0 |
| **VB** | 0 | 0.000028 | 0.000672 | 0 | 0.000028 |
| **JJ** | 0 | 0 | 0.000340 | 0.000097 | 0 |
| **NN** | 0 | 0.000200 | 0.000223 | 0.000006 | 0.002337 |
| **RB** | 0 | 0 | 0.010446 | 0 | 0 |
| **DT** | 0 | 0 | 0 | 0.506099 | 0 |

# Hidden Markov Model – Complexity

Menti.com
7544 5229

Question: What is the most likely state sequence given an output sequence?

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

- Naïve solution:
  - brute force search by enumerating all possible sequences of states
  - Complexity $O(s^m)$
    - where m is the length of the input and s is the number of states in the model.
- Better solution: Dynamic Programming!
  - Standard procedure is called the **Viterbi algorithm**
  - Running time is $O(ms^2)$,
    - where m is the length of the input and s is the number of states in the model.

# Viterbi algorithm: Motivation

Menti.com
7544 5229

- Let $T$: $o_1, o_2, \dots, o_T$ some input sentence
- Let $S$: $s_1, s_2, \dots, s_T$ sequence of tags
- Goal: best sequence of tags given the input sequence
  - $\operatorname{argmax}_{s_1, s_2, \dots, s_T} P(o_1, \dots, o_T, s_T, s_2, \dots, s_T)$
- Example
  - T = The, man, saw, the saw
  - S = {Det, N, V}
  - Possible tag sequences:

-> Det Det Det Det Det -> 0.000003
Det Det Det Det N -> 0.000008
Det Det Det Det V -> 0.000009
…
Det N V Det N -> 0.012

we have $3^5$ (243) sequences: sentence length = 5, #tags = 3

# Viterbi algorithm: Basic idea

Menti.com
7544 5229

Let us say we have only two possible states, A and B, and some observation o
What is the best possible state sequence of length 5 for this observation o?
It is either:

- the best possible sequence of length 4 that ends with A, followed by A
- the best possible sequence of length 4 that ends with A, followed by B
- the best possible sequence of length 4 that ends with B, followed by A
- the best possible sequence of length 4 that ends with B, followed by B

This is only true because the next state only depends on the state directly before!

So, what is the best possible sequence of length 4 that ends with A? It is either:

- the best possible sequence of length 3 that ends with A, followed by A
- the best possible sequence of length 3 that ends with B, followed by A
- …

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3

P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1

| | The | man | saw | the | saw | $$ |
|---|---|---|---|---|---|---|
| DT | | | | | | |
| V | | | | | | |
| N | | | | | | |

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3

P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1

| | The | man | saw | the | saw | $$ |
|---|---|---|---|---|---|---|
| DT | 0.6 * 0.5 = 0.3 | | | | | |
| V | | | | | | |
| N | | | | | | |

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | | | |
|---|---|---|---|
| P (DT \| DT) | = 0.1 | P (DT \| N) | = 0.2 |
| P (V \| DT) | = 0.3 | P (V \| N) | = 0.6 |
| P (N \| DT) | = 0.6 | P (N \| N) | = 0.2 |
| P (DT \| V) | = 0.5 | P (DT \| q0) | = 0.6 |
| P (V \| V) | = 0.2 | P (V \| q0) | = 0.3 |
| P (N \| V) | = 0.3 | P (N \| q0) | = 0.1 |

| | The | man | saw | the | saw | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | | | | | |
| V | 0 | | | | | |
| N | 0 | | | | | |

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT) = 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man |
|---|---|---|
| DT | 0.3 | |
| V | 0 | ? |
| N | 0 | |

= max {
P (man | V) * P (V | DT) * 0.3,
P (man | V) * P (V | V) * 0,
P (man | V) * P (V | N) * 0
}

Menti.com
7544 5229

# Viterbi algorithm: Example

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | | | |
|---|---|---|---|
| P (DT \| DT) | = 0.1 | P (DT \| N) | = 0.2 |
| P (V \| DT) | = 0.3 | P (V \| N) | = 0.6 |
| P (N \| DT) | = 0.6 | P (N \| N) | = 0.2 |
| P (DT \| V) | = 0.5 | P (DT \| q0) | = 0.6 |
| P (V \| V) | = 0.2 | P (V \| q0) | = 0.3 |
| P (N \| V) | = 0.3 | P (N \| q0) | = 0.1 |

| | The | man |
|---|---|---|
| DT | 0.3 | |
| V | 0 | ? |
| N | 0 | |

= max {
P (man | V) * P (V | DT) * 0.3,
P (man | V) * P (V | V) * 0,
P (man | V) * P (V | N) * 0
}

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man |
|---|---|---|
| DT | 0.3 | |
| V | 0 | ? |
| N | 0 | |

= max {
P (man | V) * P (V | DT) * 0.3,
P (man | V) * P (V | V) * 0,
P (man | V) * P (V | N) * 0
}

# Viterbi algorithm: Example

Menti.com
7544 5229

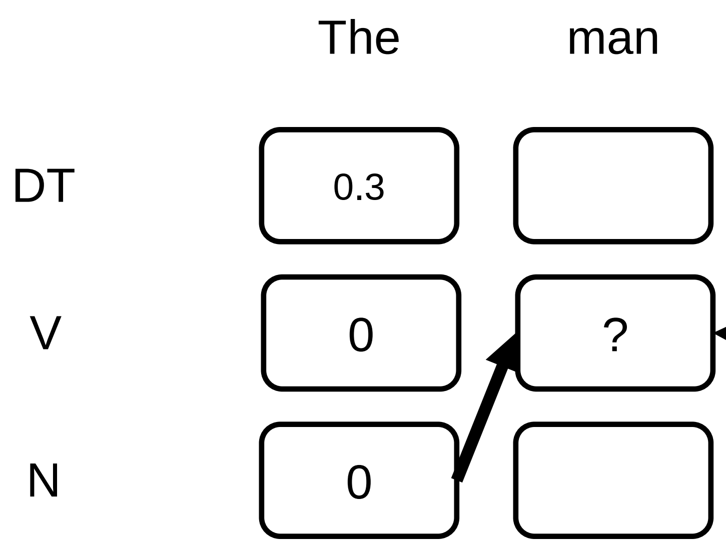
<u>Emission probabilities</u>

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

<u>Transition probabilities</u>

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3

P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1




= max {
P (man | V) * P (V | DT) * 0.3,
P (man | V) * P (V | V) * 0,
P (man | V) * P (V | N) * 0
}

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man |
|---|---|---|
| DT | 0.3 | |
| V | 0 | ? |
| N | 0 | |

= max {
0.1 * 0.3 * 0.3,
0.1 * 0.2 * 0,
0.1 * 0.6 * 0
} = 0.009

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man |
|---|---|---|
| DT | 0.3 | |
| V | 0 | 0.009<br>DT |
| N | 0 | |

= max {
0.1 * 0.3 * 0.3,
0.1 * 0.2 * 0,
0.1 * 0.6 * 0
} = 0.009

Predecessor of the most probable path

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3
P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1

| | The | man |
|---|---|---|
| DT | 0.3 | 0 |
| V | 0 | 0.009 DT |
| N | 0 | 0.054 DT |

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man | saw |
|---|---|---|---|
| DT | 0.3 | 0 | |
| V | 0 | 0.009 DT | ? |
| N | 0 | 0.054 DT | |

= max {
P (saw | V) * P (V | DT) * 0,
P (saw | V) * P (V | V) * 0.009,
P (saw | V) * P (V | N) * 0.054
}

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

| | |
|---|---|
| P (DT \| DT) = 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man | saw |
|---|---|---|---|
| DT | 0.3 | 0 | |
| V | 0 | 0.009 DT | $6{,}48 * 10^{-3}$ N |
| N | 0 | 0.054 DT | |

= max {
0.2 * 0.3 * 0, ( = 0)
0.2 * 0.2 * 0.009, ( = 0,00036)
0.2 * 0.6 * 0.054 ( = 0,00648)
} = 0,00648 = $6{,}48 * 10^{-3}$

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3
P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1

| | The | man | saw | the | saw | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | 0 | 0 | $1{,}62 * 10^{-3}$ V | 0 | |
| V | 0 | 0.009 DT | $6{,}48 * 10^{-3}$ N | 0 | $9{,}72 * 10^{-5}$ DT | $1{,}94 * 10^{-4}$ N |
| N | 0 | 0.054 DT | $2{,}16 * 10^{-3}$ N | 0 | $1{,}94 * 10^{-4}$ DT | |

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man | saw | the | saw N | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | 0 | 0 | $1{,}62 * 10^{-3}$ V | 0 | |
| V | 0 | 0.009 DT | $6{,}48 * 10^{-3}$ N | 0 | $9{,}72 * 10^{-5}$ DT | **N** |
| N | 0 | 0.054 DT | $2{,}16 * 10^{-3}$ N | 0 | $1{,}94 * 10^{-4}$ DT | |

# Viterbi algorithm: Example

Menti.com
7544 5229

**Emission probabilities**

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

**Transition probabilities**

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The | man | saw | the<br>DT | saw<br>N | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | 0 | 0 | $1{,}62 * 10^{-3}$<br>V | 0 | |
| V | 0 | 0.009<br>DT | $6{,}48 * 10^{-3}$<br>N | 0 | $9{,}72 * 10^{-5}$<br>DT | **N** |
| N | 0 | 0.054<br>DT | $2{,}16 * 10^{-3}$<br>N | 0 | **DT** | |

# Viterbi algorithm: Example

Menti.com
7544 5229

Emission probabilities

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

Transition probabilities

P (DT | DT)= 0.1
P (V | DT) = 0.3
P (N | DT) = 0.6
P (DT | V) = 0.5
P (V | V) = 0.2
P (N | V) = 0.3

P (DT | N) = 0.2
P (V | N) = 0.6
P (N | N) = 0.2
P (DT | q0) = 0.6
P (V | q0) = 0.3
P (N | q0) = 0.1

| | The | man | saw V | the DT | saw N | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | 0 | 0 | **V** | 0 | |
| V | 0 | 0.009 DT | $6{,}48 * 10^{-3}$ N | 0 | $9{,}72 * 10^{-5}$ DT | **N** |
| N | 0 | 0.054 DT | $2{,}16 * 10^{-3}$ N | 0 | **DT** | |

# Viterbi algorithm: Example

Menti.com
7544 5229

<u>Emission probabilities</u>

P (the | DT) = 0.5
P (man | V) = 0.1
P (man | N) = 0.3
P (saw | V) = 0.2
P (saw | N) = 0.2

<u>Transition probabilities</u>

| | |
|---|---|
| P (DT \| DT)= 0.1 | P (DT \| N) = 0.2 |
| P (V \| DT) = 0.3 | P (V \| N) = 0.6 |
| P (N \| DT) = 0.6 | P (N \| N) = 0.2 |
| P (DT \| V) = 0.5 | P (DT \| q0) = 0.6 |
| P (V \| V) = 0.2 | P (V \| q0) = 0.3 |
| P (N \| V) = 0.3 | P (N \| q0) = 0.1 |

| | The<br>DT | man<br>N | saw<br>V | the<br>DT | saw<br>N | $$ |
|---|---|---|---|---|---|---|
| DT | 0.3 | 0 | 0 | **V** | 0 | |
| V | 0 | 0.009<br>DT | **N** | 0 | $9{,}72 * 10^{-5}$<br>DT | **N** |
| N | 0 | **DT** | $2{,}16 * 10^{-3}$<br>N | 0 | **DT** | |

# Viterbi algorithm: Example

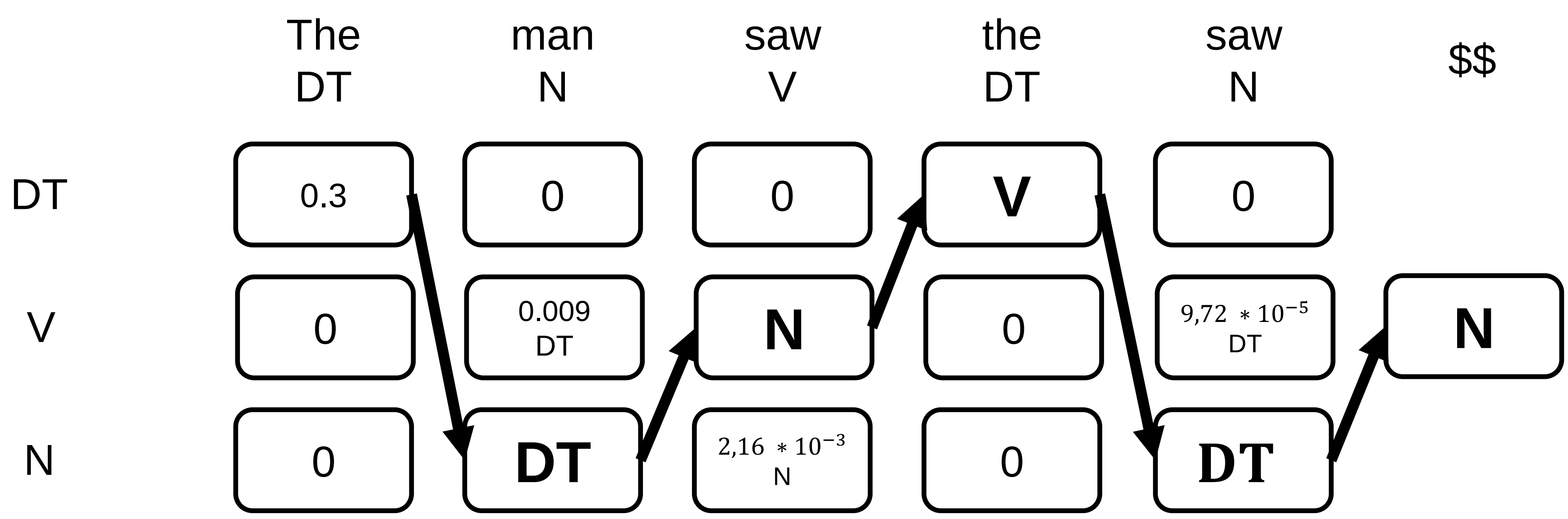
Menti.com
7544 5229

Remember the complexity of the Viterbi algorithm?

Running time is $O(ms^2)$,

- where m is the length of the input and s is the number of states in the model.

Now we see why: For every token (m) we we have to evaluate every POS (s) in combination with every possible predeccessor POS (s), so he have m * s * s operations = $ms^2$



# Summary

Menti.com
7544 5229

- Sequence Labeling:
  - Input and output are signal sequences
  - No individual classification per signal, but joint classification that minimizes some cost
- Hidden Markov Models
  - Emissions can be observed
  - States are hidden
  - Goal: Find most probable state sequence for a given emission sequence
  - Solve via Viterbi (dynamic programming)

# Next Lecture

Menti.com
7544 5229

**Information Retrieval Introduction**