

NLP

Wydział Biochemii, Biofizyki i Biotechnologii
Adrian Kania

13 listopada 2020

"The chicken is ready to eat."



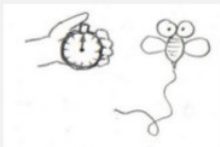
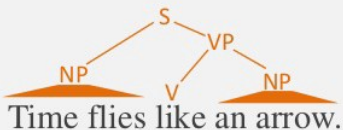
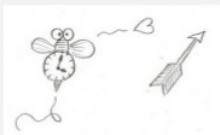
The first sentence

"They ate pizza with anchovies", can be interpreted as

- (i) "they ate pizza and the pizza had anchovies on it",
- (ii) they ate pizza using anchovies
- (iii) they ate pizza and their anchovy friends ate pizza with them.



Syntactic Ambiguity



Dane są 4 zdania, gdzie tagi każdego wyrazu są znane:

- 1 Mary Jane can see Will
- 2 Spot will see Mary
- 3 Will Jane spot Mary
- 4 Mary will pat Spot

Cel: Zbudować model HMM służący do predykcji POS tagów dla kolejnych zdań

Mary Jane can see Will

Spot will see Mary

Will Jane spot Mary

Mary will pat Spot

Mary Jane can see Will

Spot will see Mary

Will Jane spot Mary

Mary will pat Spot

	N	M	V
mary	4	0	0
jane	2	0	0
will	1	3	0
spot	2	0	1
can	0	1	0
see	0	0	2
pat	0	0	1
all	9	4	4

	N	M	V
mary	4	0	0
jane	2	0	0
will	1	3	0
spot	2	0	1
can	0	1	0
see	0	0	2
pat	0	0	1
all	9	4	4

Prawdopodobieństwa emisji:

$$P(\text{mary} | N) = ?$$

	N	M	V
mary	4	0	0
jane	2	0	0
will	1	3	0
spot	2	0	1
can	0	1	0
see	0	0	2
pat	0	0	1
all	9	4	4

Prawdopodobieństwa emisji:

$$P(\text{mary} | N) = 4/9 = 0.44$$

	N	M	V
mary	4	0	0
jane	2	0	0
will	1	3	0
spot	2	0	1
can	0	1	0
see	0	0	2
pat	0	0	1
all	9	4	4

Prawdopodobieństwa emisji:

	N	M	V
mary	0,44	0,00	0,00
jane	0,22	0,00	0,00
will	0,11	0,75	0,00
spot	0,22	0,00	0,25
can	0,00	0,25	0,00
see	0,00	0,00	0,50
pat	0,00	0,00	0,25



<S>Mary Jane can see Will<E>

<S>Spot will see Mary<E>

<S>Will Jane spot Mary<E>

<S>Mary will pat Spot<E>

<S> Mary Jane can see Will <E>

<S> Spot will see Mary <E>

<S> Will Jane spot Mary <E>

<S> Mary will pat Spot <E>

	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

Prawdopodobieństwa przejść:

$$P(N|<S>) = ?$$

	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

Prawdopodobieństwa przejść:

$$P(N|<S>) = \frac{3}{4} = 0.75$$

	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

Prawdopodobieństwa przejść:

	N	M	V	<E>
<S>	0,75	0,25	0,00	0,00
N	0,11	0,33	0,11	0,44
M	0,25	0,00	0,75	0,00
V	1,00	0,00	0,00	0,00

Nowe zdanie:

Will can spot Mary

Cel:

otagować każde słowo

Prawdopodobieństwa emisji:

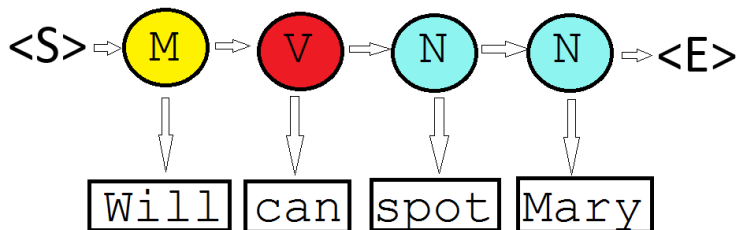
	N	M	V
mary	0,44	0,00	0,00
jane	0,22	0,00	0,00
will	0,11	0,75	0,00
spot	0,22	0,00	0,25
can	0,00	0,25	0,00
see	0,00	0,00	0,50

Prawdopodobieństwa przejść:

	N	M	V	<E>
<S>	0,75	0,25	0,00	0,00
N	0,11	0,33	0,11	0,44
M	0,25	0,00	0,75	0,00
V	1,00	0,00	0,00	0,00

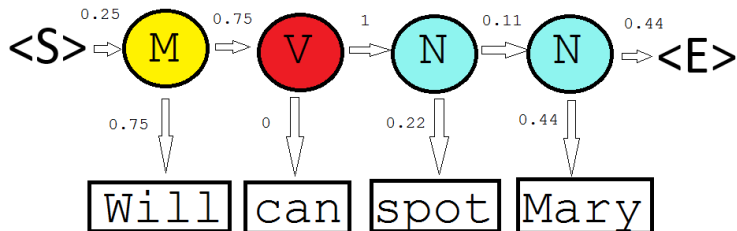
<S>Will can spot Mary<E>

<S>Will can spot Mary<E>



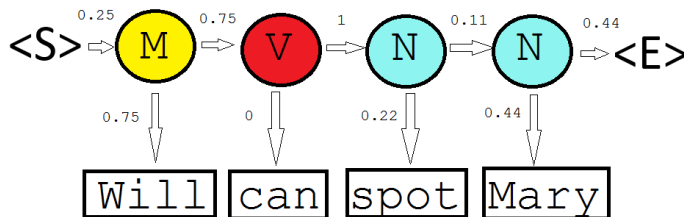
HMM - przykładowe otagowanie 1

<S>Will can spot Mary<E>



HMM - przykładowe otagowanie 1

<S> Will can spot Mary <E>

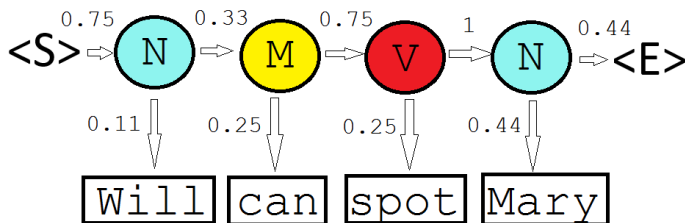


Prawdopodobieństwo poprawności takiego otagowania:

$$0.25 * 0.75 * 0.75 * 0 * 1 * 0.22 * 0.11 * 0.44 * 0.44 = 0$$

HMM - przykładowe otagowanie 2

<S> Will can spot Mary <E>



Prawdopodobieństwo poprawności takiego otagowania:
 $0.75 \cdot 0.11 \cdot 0.33 \cdot 0.25 \cdot 0.75 \cdot 0.25 \cdot 1 \cdot 0.44 \cdot 0.44 = 0.00025$

Ile jest wszystkich możliwości utworzenia zestawu tagów dla podanego zdania?

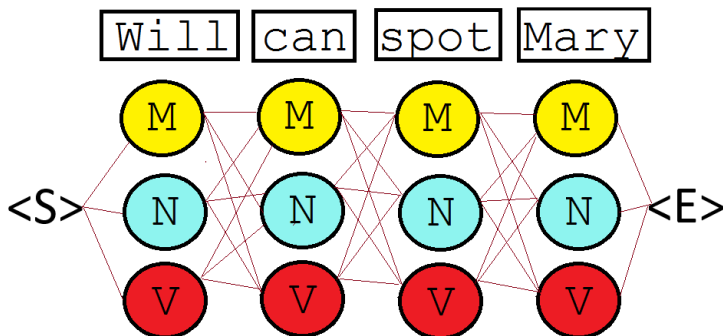
Ile jest wszystkich możliwości utworzenia zestawu tagów dla podanego zdania?

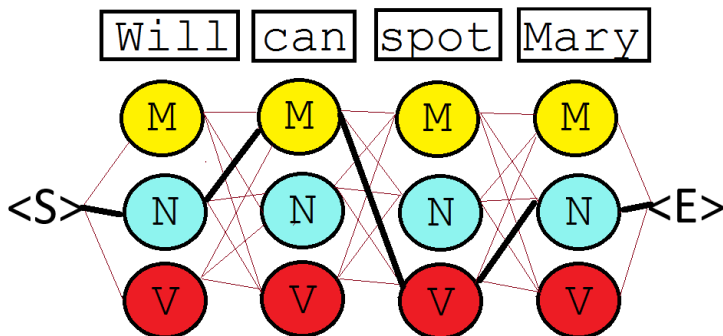
Zatem złożność $O(K^N)$, gdzie K - liczba tagów, N - liczba wyrazów.

Zatem złożoność $O(K^N)$, gdzie K - liczba tagów, N - liczba wyrazów.
 Przykładowo dla $K = 10$ i $N = 100$ mamy:

```
10000000000000 0000000000000 0000000000000 0000000000000
0000000000000 0000000000000 0000000000000 0000000000000
0000000000000 0000000000000
```

możliwości.



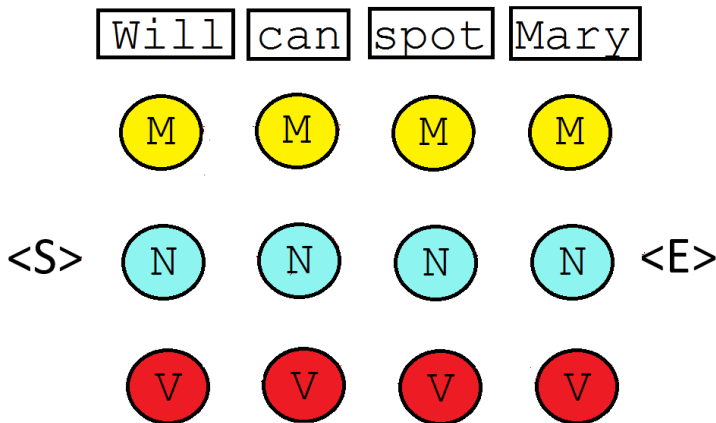


Jak znaleźć szybko najbardziej optymalną ścieżkę po stanach ukrytych?

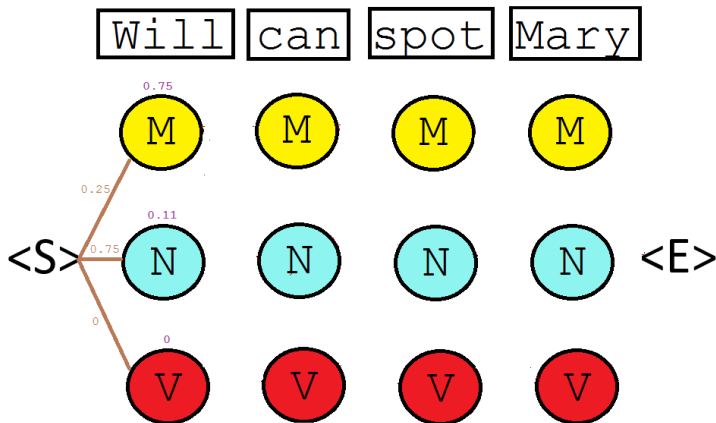
Algorytm Viterbiego

- 1 Pozwala on wskazać taką ścieżkę (a więc takie otagowanie wyrazów) której prawdopodobieństwo wystąpienia jest największe (metoda Największej Wiarygodności).
- 2 W tym przypadku złożoność wynosi $O(NK^2)$. Dla $K = 10$ i $N = 100$ daje to maksymalnie 10000 kroków.
- 3 Dlaczego maksymalnie? O tym za chwilę...
- 4 Idea bardzo prosta - nie liczymy za każdym razem iloczynu wszystkich prawdopodobieństw emisji/przejęć lecz przechowujemy informacje o tym co było wcześniej i idziemy iteracyjnie naprzód.

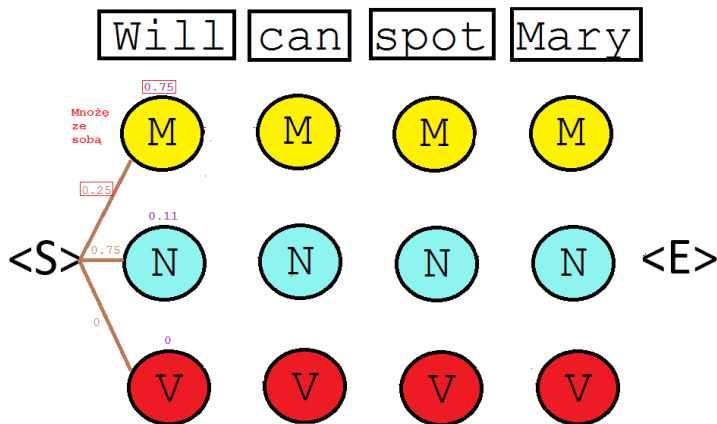
Algorytm Viterbiego



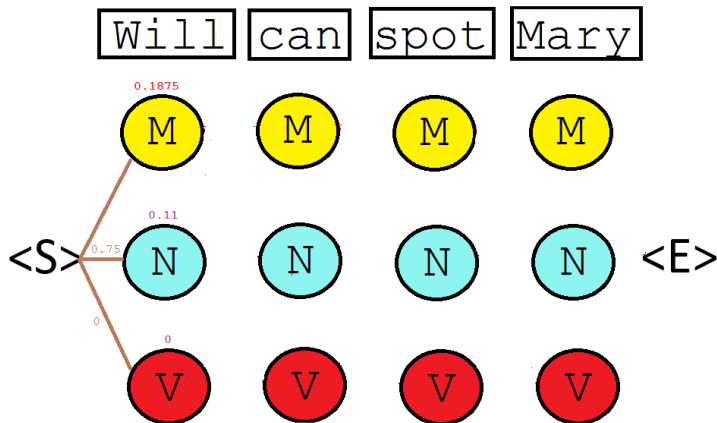
Algorytm Viterbiego



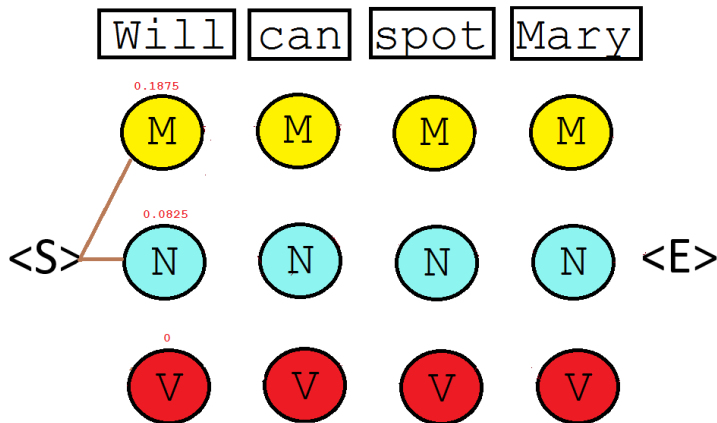
Algorytm Viterbiego



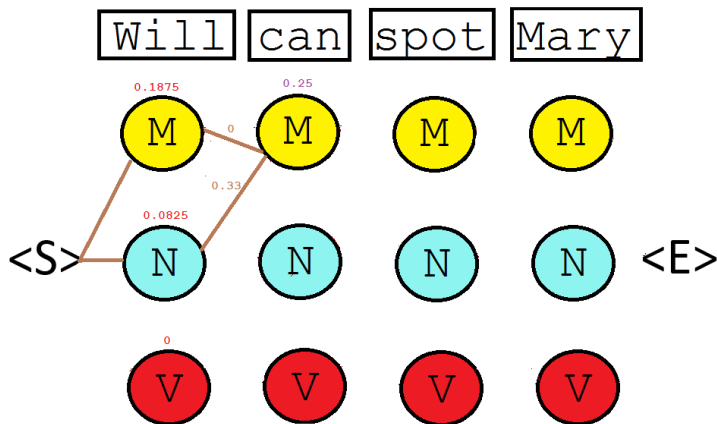
Algorytm Viterbiego



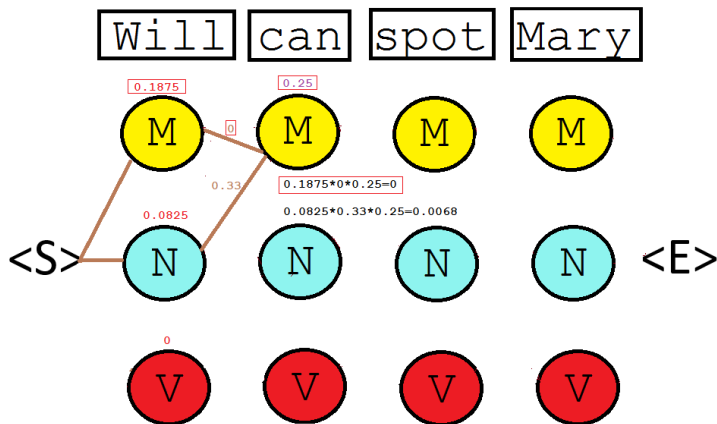
Algorytm Viterbiego



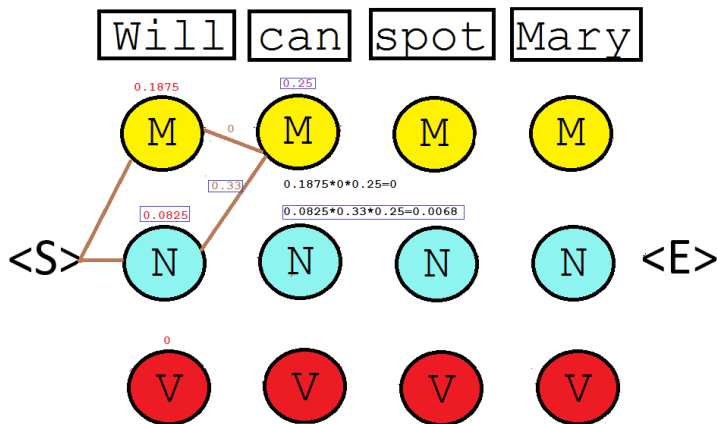
Algorytm Viterbiego



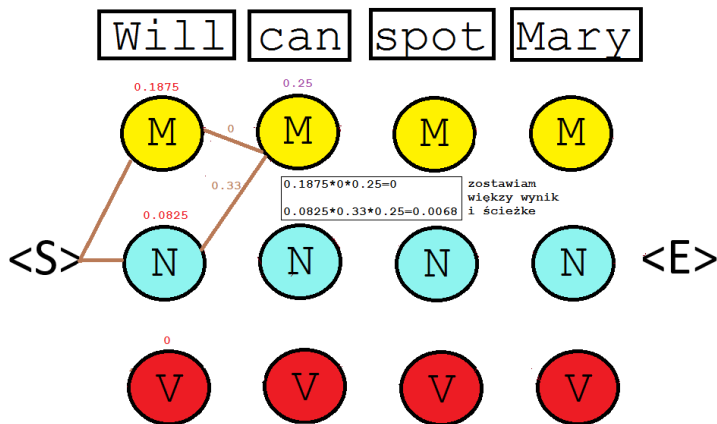
Algorytm Viterbiego



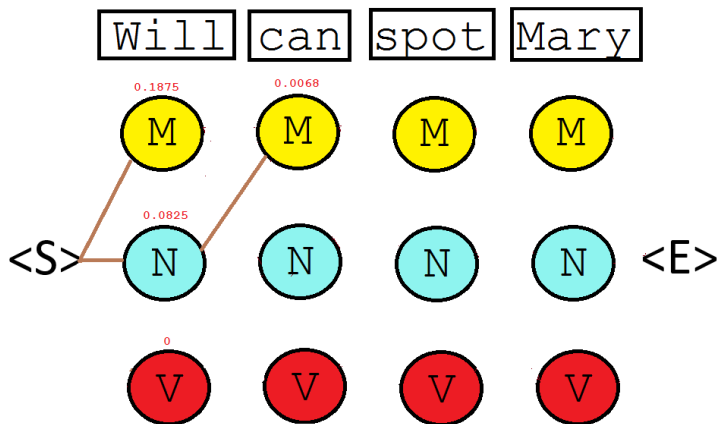
Algorytm Viterbiego



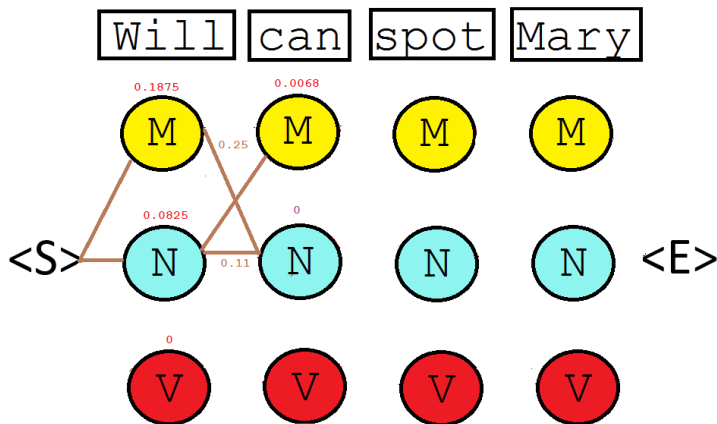
Algorytm Viterbiego



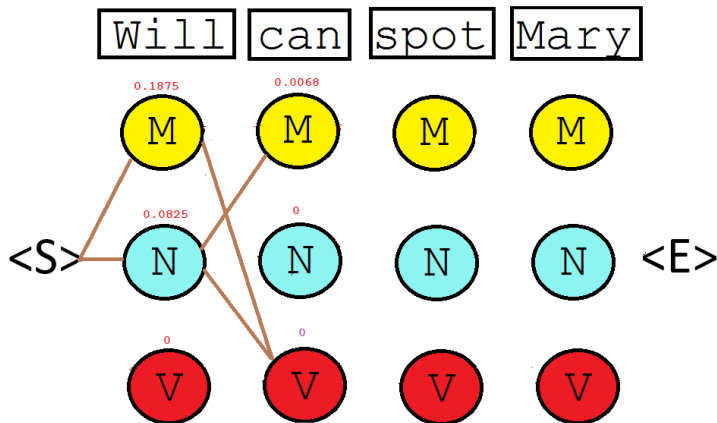
Algorytm Viterbiego



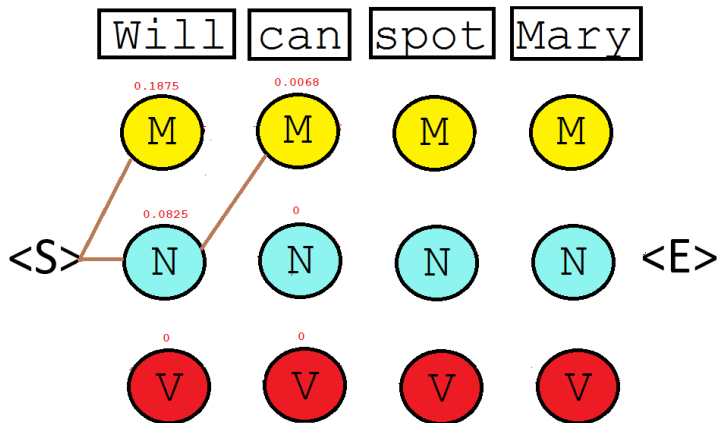
Algorytm Viterbiego



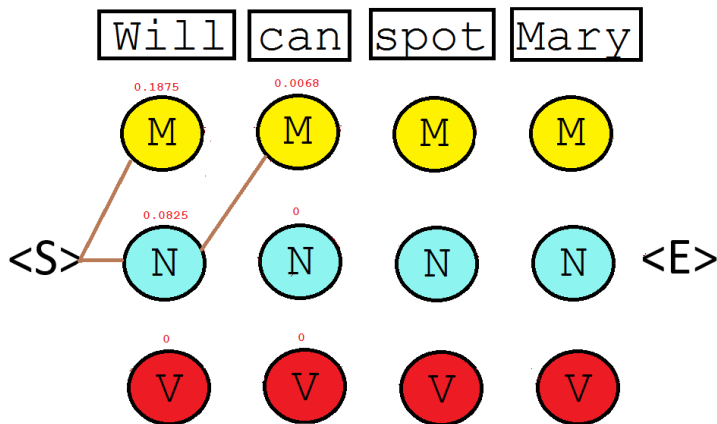
Algorytm Viterbiego



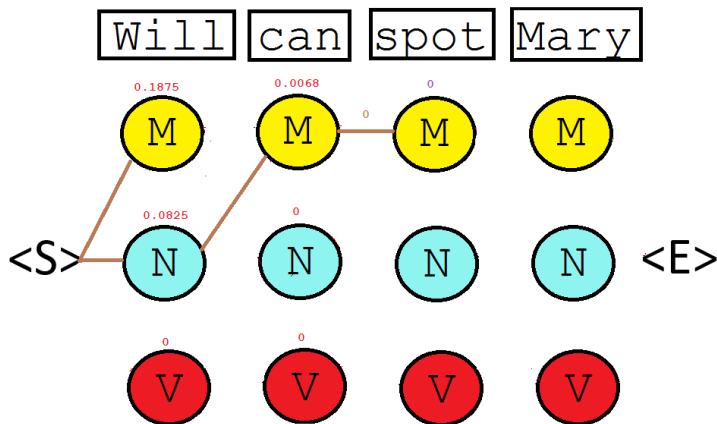
Algorytm Viterbiego



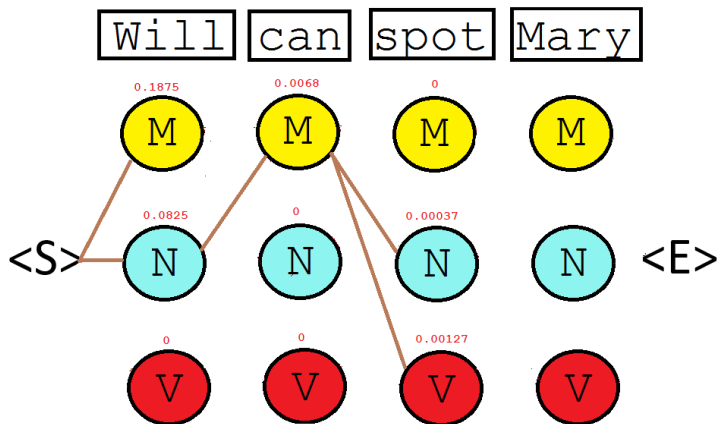
Algorytm Viterbiego



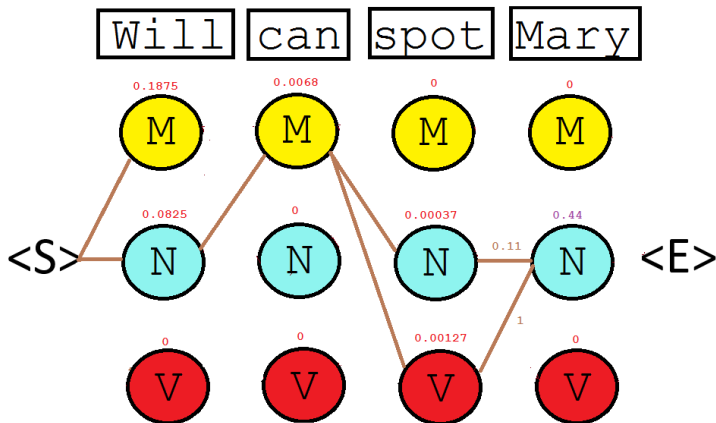
Algorytm Viterbiego



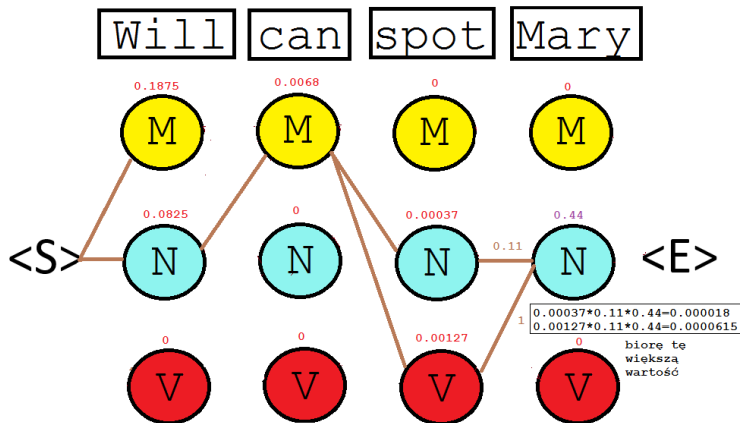
Algorytm Viterbiego



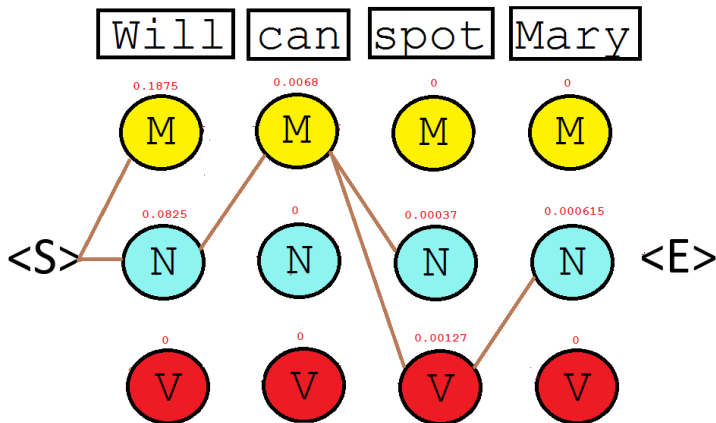
Algorytm Viterbiego



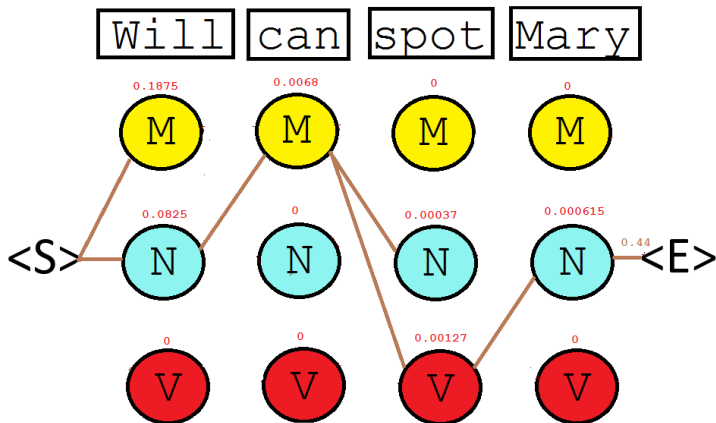
Algorytm Viterbiego



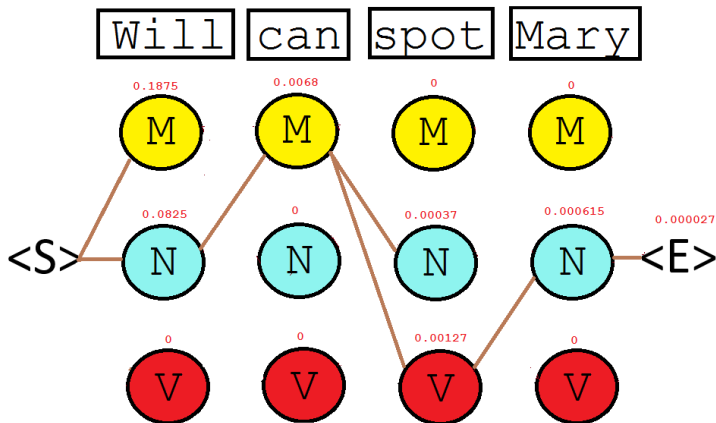
Algorytm Viterbiego



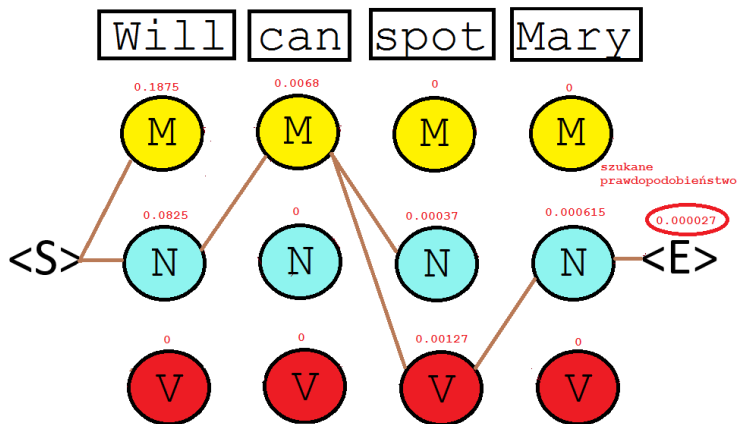
Algorytm Viterbiego



Algorytm Viterbiego

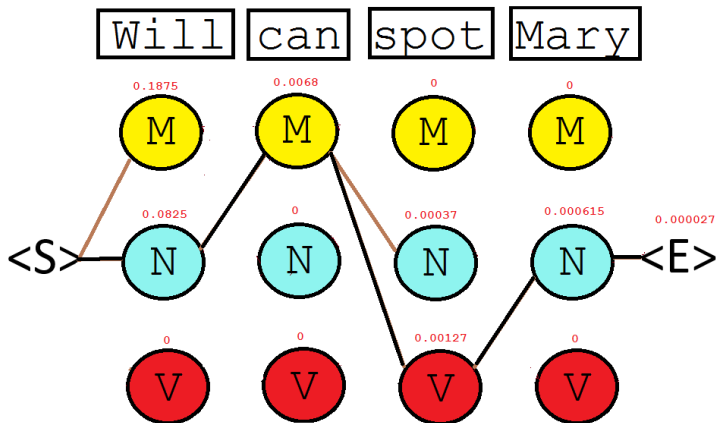


Algorytm Viterbiego



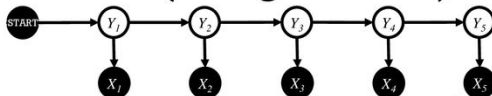
Algorytm Viterbiego

Optymalną sekwencję stanów ukrytych uzyskuje idąc od końca i przechodząc tagi z największymi prawdopodobieństwami (połączone krawędzią).

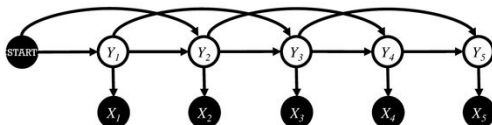


HMM - łańcuchy wyższych rzędów

1st-order HMM (i.e. bigram HMM)



2nd-order HMM (i.e. trigram HMM)



3rd-order HMM

