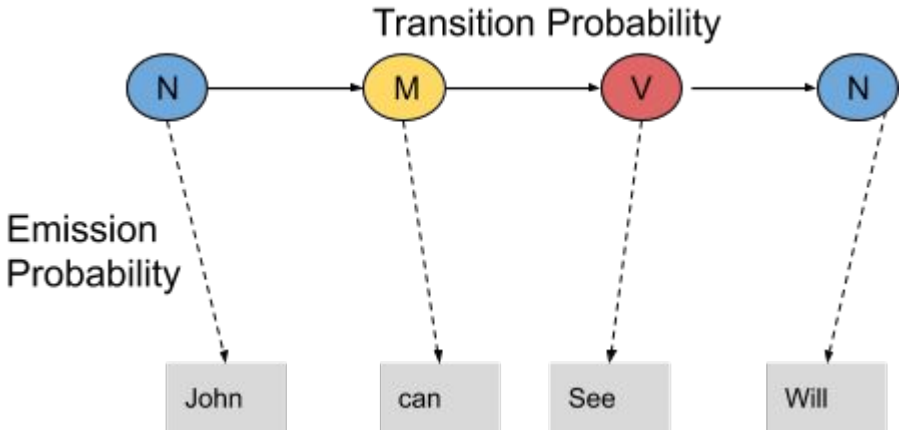
POS Tagging

HMM example, NLTK Discussion

HMM Tagger Example



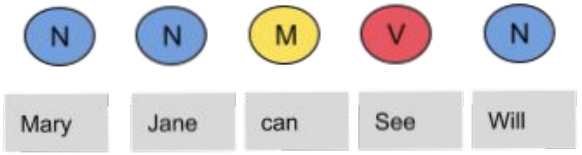
3 Tags: N, M, V (noun, model, verb)

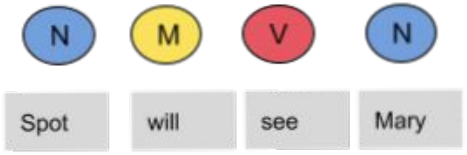
The transition probability is the likelihood of a particular sequence

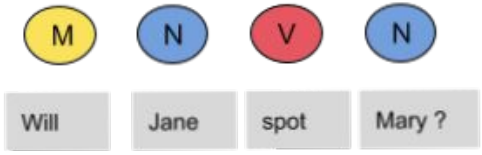
Emission probabilities are probability that a word assigned particular tag

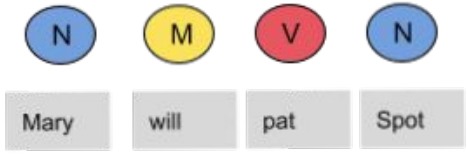
HMM Tagger Example

- Mary Jane can see Will
- Spot will see Mary
- Will Jane spot Mary?
- Mary will pat Spot

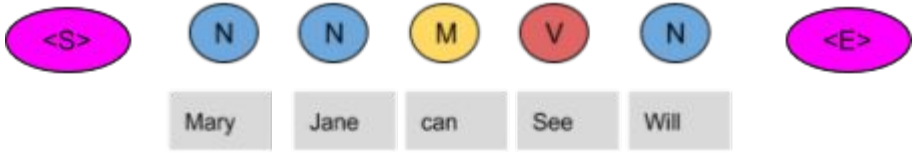




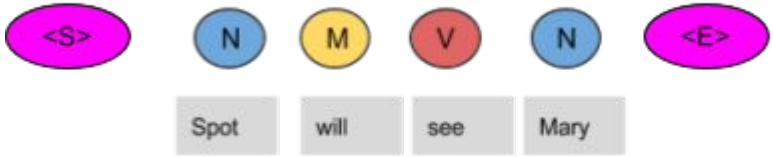


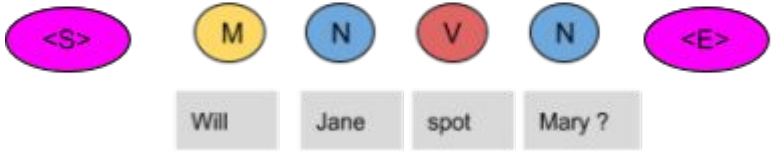


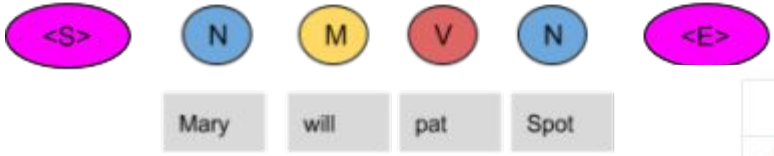
Words	Noun	Model	Verb
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
Can	0	1/4	0
See	0	0	2/4
pat	0	0	1



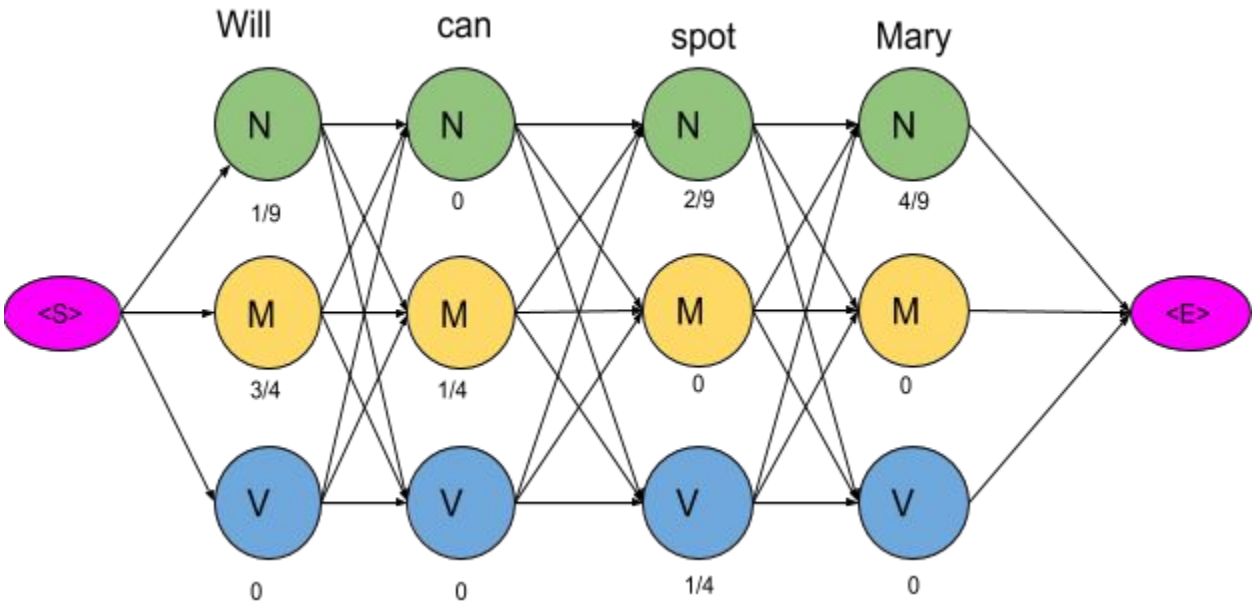
	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

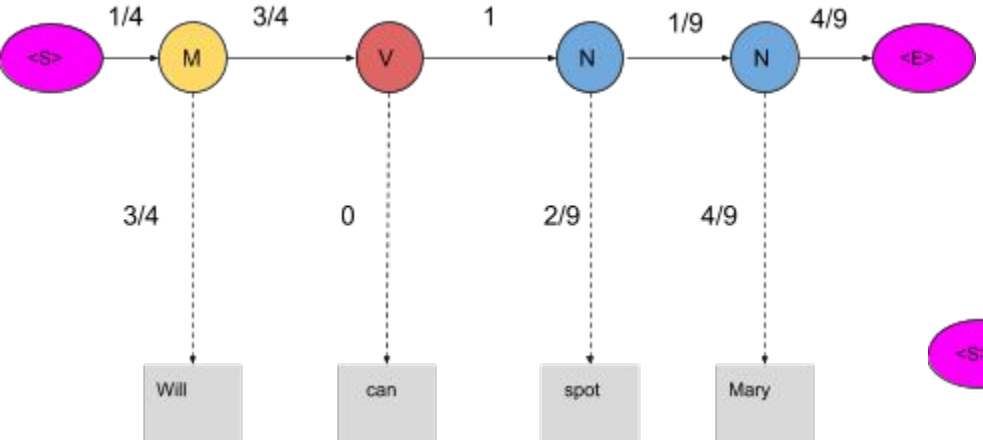




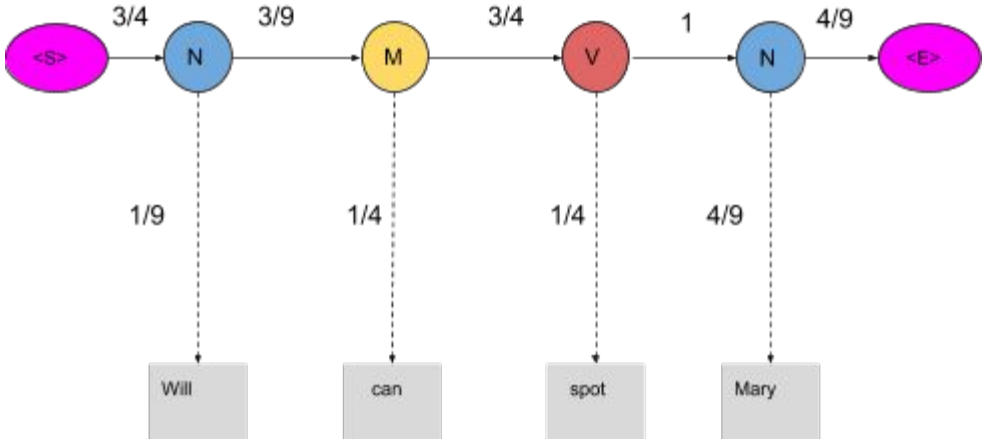


	N	M	V	<E>
<S>	3/4	1/4	0	0
N	1/9	3/9	1/9	4/9
M	1/4	0	3/4	0
V	4/4	0	0	0

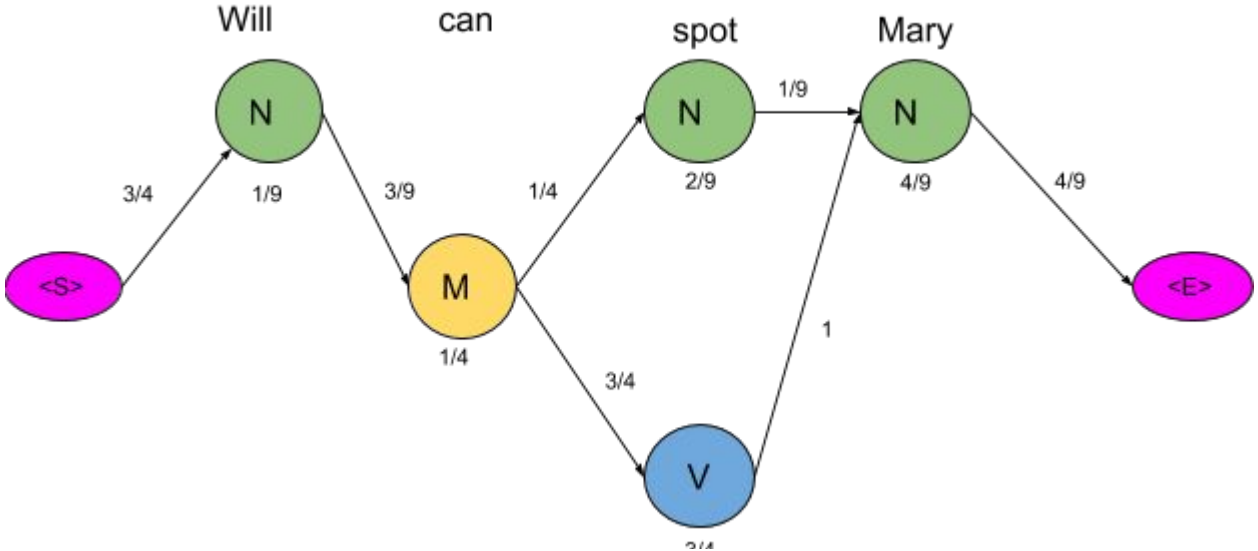

Will
can
spot
Mary
N
N
N
N
1/9
0
2/9
4/9
<S>
M
M
M
M
<E>
3/4
1/4
0
0
V
V
V
V
0
0
1/4
0



1/4*3/4*3/4*0*1*2/9*1/9*4/9*4/9=0



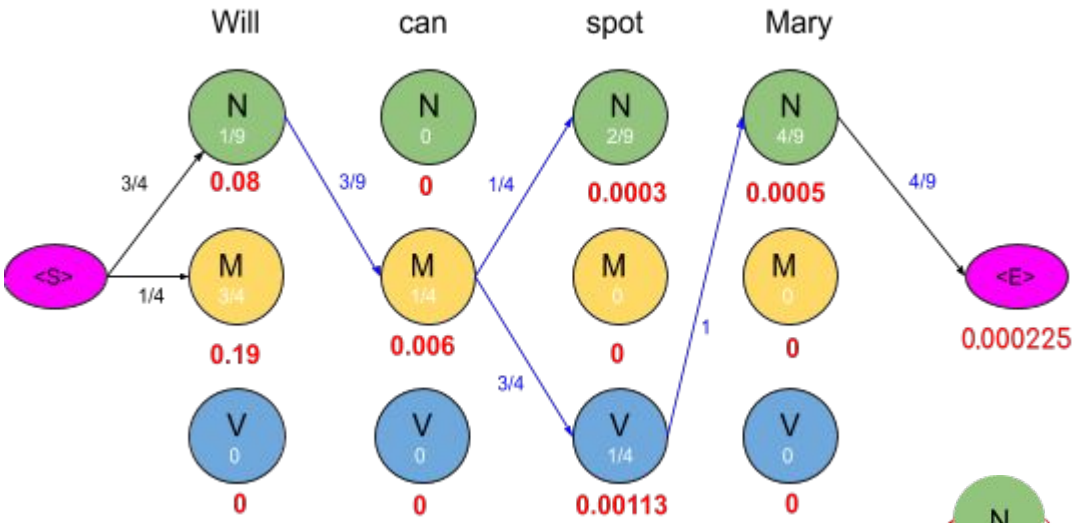
3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

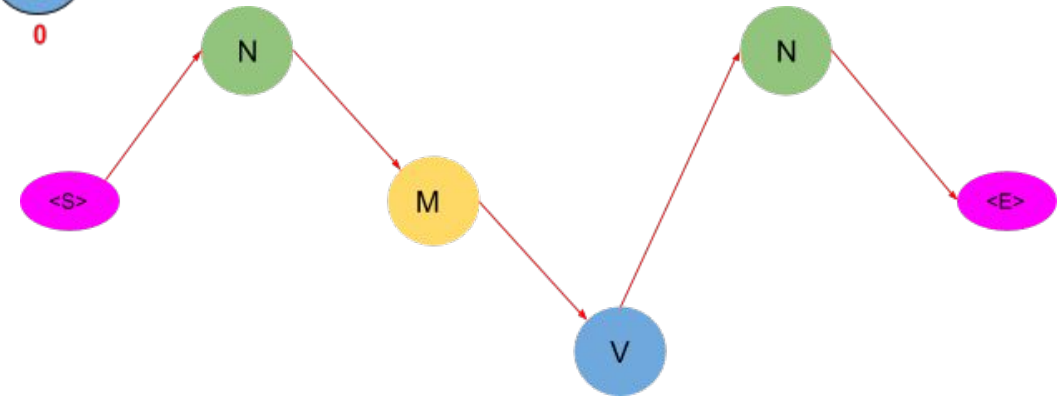


<S>→N→M→N→N→<E> =3/4*1/9*3/9*1/4*1/4*2/9*1/9*4/9*4/9=0.00000846754

<S>→N→M→N→V→<E>=3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

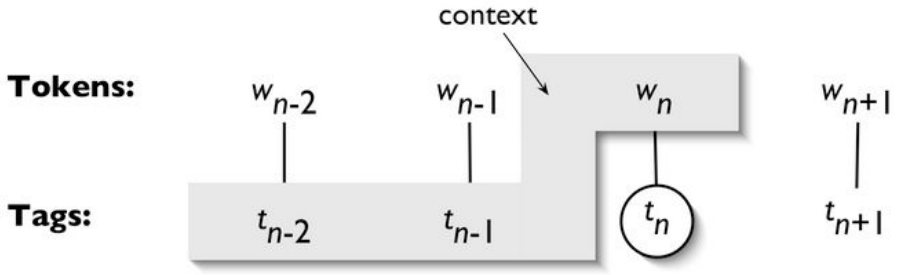
Viterbi algorithm (dynamic algorithm) provide an efficient computation





Tagging- using NLTK

- Unigram Tagging
- N-gram Tagging



- Maximum Entropy Markov Model based Tagging (MEMM)
- Conditional Random Fields
- Transformation Based Tagging

NLTK- Transformation based tagging

```
>>> from nltk.tbl import demo as brill_demo
>>> brill_demo.demo()
Training Brill tagger on 80 sentences...
Finding initial useful rules...
    Found 6555 useful rules.

           B      |
   S   F   r   O  |        Score = Fixed - Broken
   c   i   o   t  |  R     Fixed = num tags changed incorrect -> correct
   o   x   k   h  |  u     Broken = num tags changed correct -> incorrect
   r   e   e   e  |  l     Other = num tags changed incorrect -> incorrect
   e   d   n   r  |  e
------------------+-------------------------------------------------------
  12  13   1   4  | NN -> VB if the tag of the preceding word is 'TO'
   8   9   1  23  | NN -> VBD if the tag of the following word is 'DT'
   8   8   0   9  | NN -> VBD if the tag of the preceding word is 'NNS'
   6   9   3  16  | NN -> NNP if the tag of words i-2...i-1 is '-NONE-'
   5   8   3   6  | NN -> NNP if the tag of the following word is 'NNP'
   5   6   1   0  | NN -> NNP if the text of words i-2...i-1 is 'like'
   5   5   0   3  | NN -> VBN if the text of the following word is '*-1'
```

Excercise

Write programs to process the Brown Corpus and find answers to the following questions:

1. Which nouns are more common in their plural form, rather than their singular form? (Only consider regular plurals, formed with the -s suffix.)
2. Which word has the greatest number of distinct tags. What are they, and what do they represent?
3. List tags in order of decreasing frequency. What do the 20 most frequent tags represent?
4. Which tags are nouns most commonly found after? What do these tags represent?

Summary

- Words can be grouped into classes, such as nouns, verbs, adjectives, and adverbs. These classes are known as lexical categories or parts of speech. Parts of speech are assigned short labels, or tags, such as NN, VB,
- The process of automatically assigning parts of speech to words in text is called part-of-speech tagging, POS tagging, or just tagging.
- Automatic tagging is an important step in the NLP pipeline, and is useful in a variety of situations including: predicting the behavior of previously unseen words, analyzing word usage in corpora, and text-to-speech systems.
- Some linguistic corpora, such as the Brown Corpus, have been POS tagged.
- A variety of tagging methods are possible, e.g. default tagger, regular expression tagger, unigram tagger and n-gram taggers. These can be combined using a technique known as backoff.

Summary

- Taggers can be trained and evaluated using tagged corpora.
- Part-of-speech tagging is an important, early example of a sequence classification task in NLP: a classification decision at any one point in the sequence makes use of words and tags in the local context.
- A dictionary is used to map between arbitrary types of information, such as a string and a number: freq['cat'] = 12. We create dictionaries using the brace notation: pos = {}, pos = {'furiously': 'adv', 'ideas': 'n', 'colorless': 'adj'}.
- N-gram taggers can be defined for large values of n, but once n is larger than 3 we usually encounter the sparse data problem; even with a large quantity of training data we only see a tiny fraction of possible contexts.
- Transformation-based tagging involves learning a series of repair rules of the form "change tag s to tag t in context c", where each rule fixes mistakes and possibly introduces a (smaller) number of errors.