

Name:

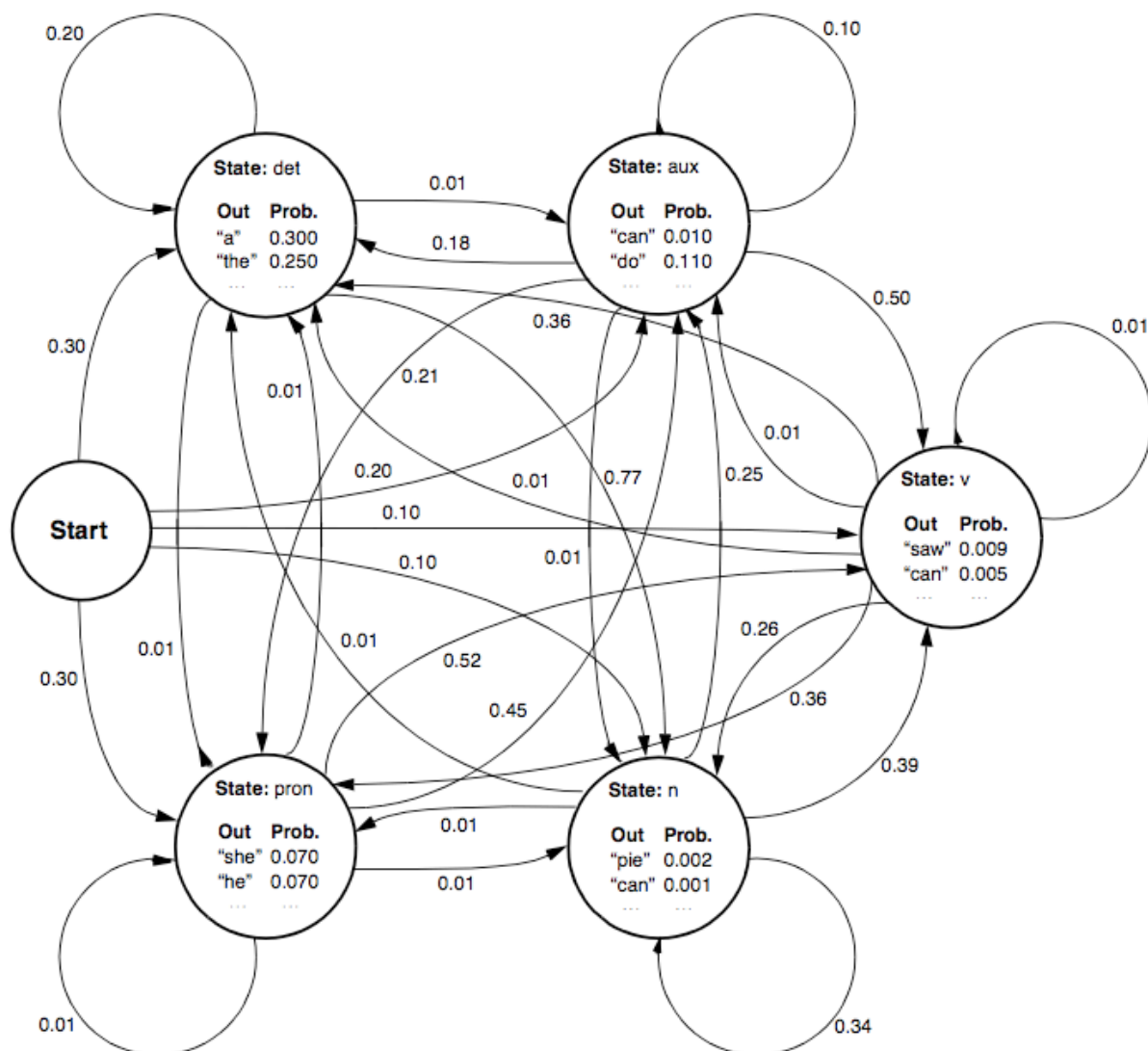
Statistical Lexical Category Disambiguation

Tutorial 4

The Parser Files

- We'll begin by looking at two simple tagger implementations in Prolog:
`/proj/courses/comppsy/Tutorial4/tagger.pl`
`/proj/courses/comppsy/Tutorial4/viterbi.pl`
`/proj/courses/comppsy/Tutorial4/viterbi2.pl`
- Copy all the files to your own directory.

The HMM model that is included with the prolog tagger corresponds to the following:



Notes: The probabilities of all transition arcs leaving a state should sum to 1.0. Similarly, the outputs from a state should also sum to 1.0, but here they don't since the lexicon is clearly incomplete.

1. Tagging in Prolog

The basic tagger (`tager.pl`) computes the probabilities of all possible paths, in parallel. There is a “write” statement which displays all the alternatives it considers (note: the tags are in reverse order). To run the tagger on a string of words (note, it doesn’t need to be a grammatical string) simply type:

```
?- most_probable_sequence([a,can,can],S).
```

```
?- most_probable_sequence([he,can,can,a,can],S).
```

Question: How many paths are considered for the second sentence? Explain why the algorithm considers exactly this many paths.

2. Viterbi in Prolog

The Viterbi algorithm (`viterbi.pl`) is much more efficient (both in time and space), because it only computes the best path, and thus avoids keeping track of any paths which cannot lead to the best path. You can try it again with the same sentences:

```
?- most_probable_sequence([a,can,can],S).
```

```
?- most_probable_sequence([he,can,can,a,can],S).
```

Question: How many paths are considered for the second sentence? Can you explain why the algorithm considers exactly this many paths. Can you tell which paths are not considered?

3. Viterbi in Prolog

Recall the Viterbi algorithm for category disambiguation, which is much more efficient, because it only computes the best path, and thus avoids keeping track of any paths which cannot lead to the best path. The HMM in `viterbi2.pl` has been modified slightly to distinguish transitive (`vt`) and intransitive verbs (`vi`), and allow a simply adverbial to appear after the verbs (but see the probabilities). The aux state has been removed.

- The man fought yesterday.
- The man held the man.

You can try it with the following sentence:

```
?- mps([the,man,fought,the,man],S).
```

a) Show the best POS sequence, and it’s probability, for both sentences above: