Tagging: Why is there a search problem?

\[ T^* = \arg \max_T P(T|W) = \arg \max_T P(W|T)P(T) \]

- Number of tag sequences for a sentence of length \( n \) is \( O(T^n) \) where \( T \) is the size of the tagset

- OK, but why is there a non-trivial search problem?
  - e.g. for a unigram model we can just take the most probable tag for each word, an algorithm which runs in \( O(nT) \) time
Tagging: Why is there a non-trivial search problem?

\[ T^* = \arg \max_T P(T | W) = \arg \max_T P(W | T) P(T) \]

- But what about a bigram model?
- Intuition: suppose I have two competing tags for word \( w_i, t^1_i \) and \( t^2_i \)
- Compare:

\[
\begin{align*}
\text{Score}(t^1_i) &= P(t^1_i | t_{i-1}) P(w_i | t^1_i)) \\
\text{Score}(t^2_i) &= P(t^2_i | t_{i-1}) P(w_i | t^2_i))
\end{align*}
\]

Suppose \( \text{Score}(t^1_i) > \text{Score}(t^2_i) \); can we be sure \( t^1_i \) is part of the highest scoring tag sequence?
Viterbi Algorithm

- Dynamic Programming (DP) algorithm, so requires the “optimal sub-problem property”
  - i.e. optimal solution to the complete problem can be defined in terms of optimal solutions to sub-problems
- So what are the sub-problems in this case?
  - intuition: suppose we want the optimal tag sequence ending at $w_n$, and we know the optimal sub-sequence ending at $w_{n-1}$, for all possible tags at $w_{n-1}$
Viterbi Algorithm (for a bigram tagger)

\[
\delta_{t_j}(n + 1) = \max_{t_i} \delta_{t_i}(n) P(t_i|t_j) P(w_i|t_i)
\]

where \( \delta_{t_j}(n+1) \) is the probability of the most probable tag sequence ending in tag \( t_j \) at position \( n + 1 \)

- Recursion bottoms out at position 1 in the sentence
- Most probable tag sequence can be obtained by following the recursion from the right backwards
- Time complexity is \( O(T^2n) \) where \( T \) is the size of the tagset

[See Chs. 9 and 10 of Manning and Schutze for a more rigorous presentation]
Practicalities

• Choice of tags to be assigned to a particular word usually governed by a “tag dictionary”

• Accuracy measured by taking a manually created “gold-standard” for a set of held-out test sentences

• Accuracy of POS taggers on newspaper data is over 97%, close to the upper bound represented by human agreement (and existence of noise in the data)

• Linear time process (in length of sentence) means tagging can be performed very fast, e.g. hundreds of thousands of words per second
It is difficult to understand what I want to do.

Taken from Dienes(2004)
• More direct representation of how the words in a sentence are related, in terms of (labelled) edges between words

• Currently a popular form of parsing:
  – interesting algorithmic and learning problems;
  – useful for applications;
  – applicable to all languages (including eg free word order languages)
  – theory-neutral (to a large extent)
• What is the grammar of the natural language in question? Where does it come from?

• What is the algorithm which builds the possible parses for a sentence?

• What is the model for determining the plausibility of the parses (because there may be lots of alternatives)?
Why is Parsing Difficult?

- Obtaining a *wide-coverage* grammar which can handle arbitrary real text is challenging
- Natural language is surprisingly *ambiguous*
Syntactic Ambiguity
Syntactic Ambiguity: the problem is worse than you think
Syntactic Ambiguity: the problem is worse than you think
Syntactic Ambiguity: the problem is even worse than that

• Put the block in the box on the table 2 analyses
• Put the block in the box on the table beside the chair 5 analyses
• Put the block in the box on the table beside the chair before the table 14 analyses
• Put the block in the box on the table beside the chair before the table in the kitchen 42 analyses
• ... 132 analyses
• ... 469 analyses
• ... 1430 analyses
• ... 4862 analyses
• Previous sequence was the Catalan sequence; grows exponentially with the number of PPs

• Question: Ok, but we never see PPs stacked up like that in real sentences?

• Answer: but we do see other constructions with similar behaviour, eg coordination, and these various constructions stack up against each other
Syntactic Ambiguity: the problem is even worse than that

- Wider grammar coverage $\Rightarrow$ more analyses
- In practice this could mean millions (or more) of parses for a single sentence
  - difficult to imagine how productive these wide-coverage grammars can be without looking carefully at the output of a parser which uses one
- We need an efficient representation of this parse space
- And an efficient way to search it

[show crazy analysis from online demo]
• Chapters 9 and 10 of Manning and Schutze