Astar heuristic for POS Tagging - Bigram

Jayaprakash
Nikhil
Subhash

Goal

$$P(T/W) = \underset{T}{\operatorname{argmax}} \left(P(w_1/t_1) * P(t_2/t_1) * P(w_2/t_2) * P(t_3/t_2) * \ldots\ldots P(w_n/t_n) * P(t_{n+1}/t_n) \right)$$

After taking the log and negating it, reduces to minimization problem

$$P(T/W) = \underset{T}{\operatorname{argmin}} \left(LP(w_1/t_1) + LP(t_2/t_1) + LP(w_2/t_2) + LP(t_3/t_2) + \ldots\ldots + LP(w_n/t_n) + LP(t_{n+1}/t_n) \right)$$

Optimal path has $min\left(\Sigma - \log\left(P\left(S_i \rightarrow S_{(i+1)}\right)\right)\right)$

Heuristics

h(n) = minimum cost path from n to G (without considering emission probability)+ minimum emission probability at each level

Admissible ?

$LP(w_1/t_1) + LP(t_2/t_1) +$
$LP(w_2/t_2) + LP(t_3/t_2) +$
.....
$LP(w_n/t_n) + LP(t_{n+1}/t_n)$

>=

$\min_T (LP(t_2/t_1) + LP(t_3/t_2) \ldots\ldots LP(t_{n+1}/t_n))$

+

$\min_t(LP(w_1/t)) + \min_t(LP(w_2/t)) \ldots \min_t (LP(w_n/t))$

Code

```
public double get_h(int heuristic) {
  double res=0.0;

  //min path  to goal with length goal.depth-this.depth-1
  res = res + AStar.minState[goal.depth-this.depth-1][this.state];

  //sum of min emiss probs from  wk to wn
  res = res + AStar.minEmis[this.depth+1];

  return res;
}
```

Calculation ?

$$\min_{T} \left(LP(t_2/t_1) + LP(t_3/t_2) \ldots\ldots LP(t_{n+1}/t_n) \right)$$

Since this term does not depends on the given sentence it can be calculated once and reused in constant time.

+

$$\min_{t}(LP(w_1/t)) + \min_{t}(LP(w_1/t)) \ldots \min_{t}(LP(w_n/t))$$

This is calculated once per entire sentence.

Results

* Reduced the open nodes count **about 18 %**
* ~ 37 Lacs nodes were opened basic heuristic
* ~ 30 Lacs nodes were opened with the this heuristics
* Total words = 935969
  Total sentences = 41660
  Avg sentence length = 22.46