Integrating Beam Search and Error States

Group 6

What we have done

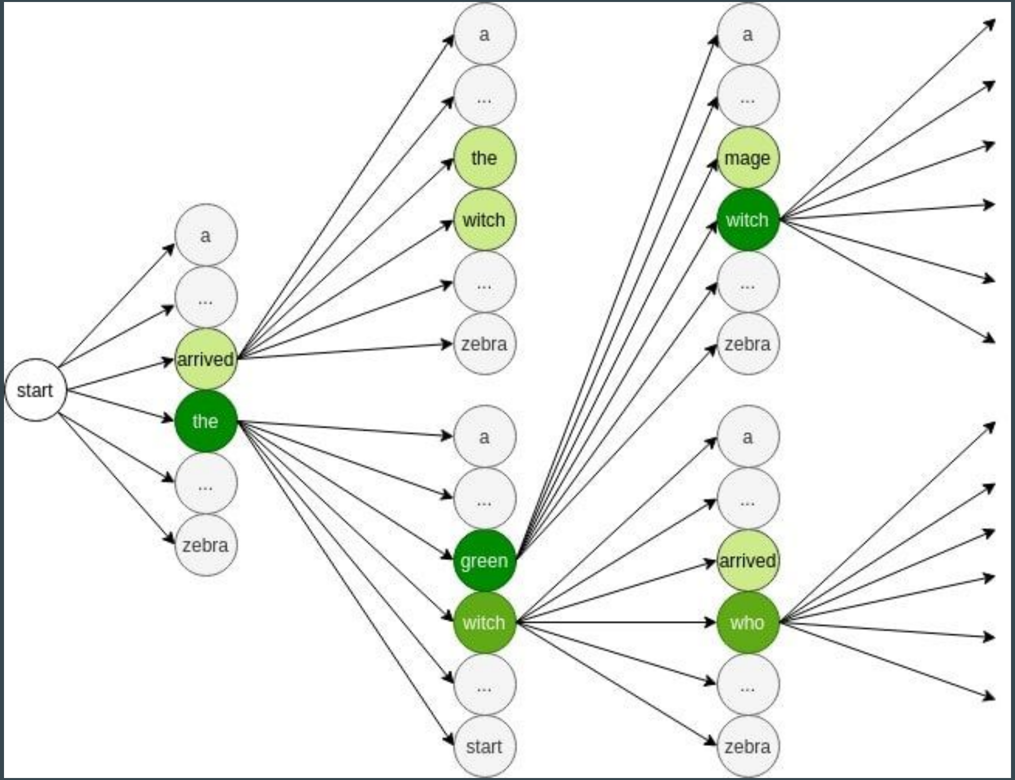
- Extension from baseline project
- Exchange the greedy search to a beam search
- Best first beam search.
- Introduce error states.

Why is this interesting?

- Beam search will improve accuracy
- Best first implementation can efficiency without at the cost of accuracy.
- Learn from its mistakes.

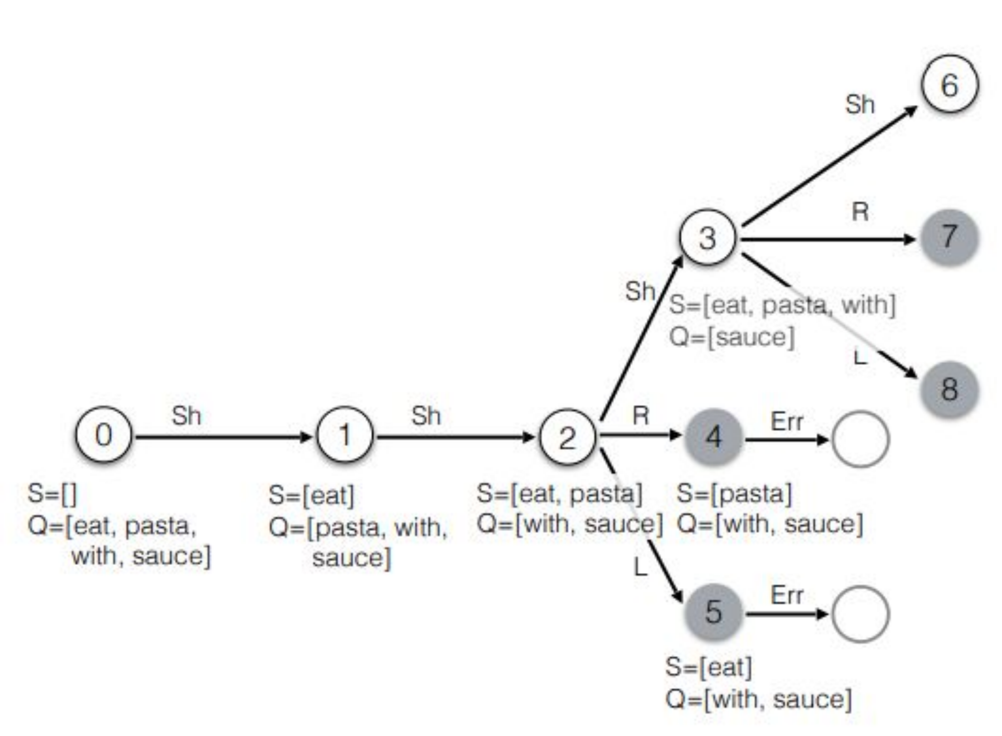
BEAM SEARCH

- Efficient Search Algorithm
- Maintains Beam of Top Hypotheses
- Expands Promising Hypotheses
- Prunes Unlikely Hypotheses



ERROR STATES

- Address parsing mistakes
- Improve accuracy
- Recover from mistakes



Related work

If beam search is the answer, what was the question?

Investigated beam search

Try to answer why it is effective

Analyzed impact of beam widths

Best-First Beam Search

Proposed best-first beam search

Prioritizes the highest scoring paths at all times

Similar accuracy but faster

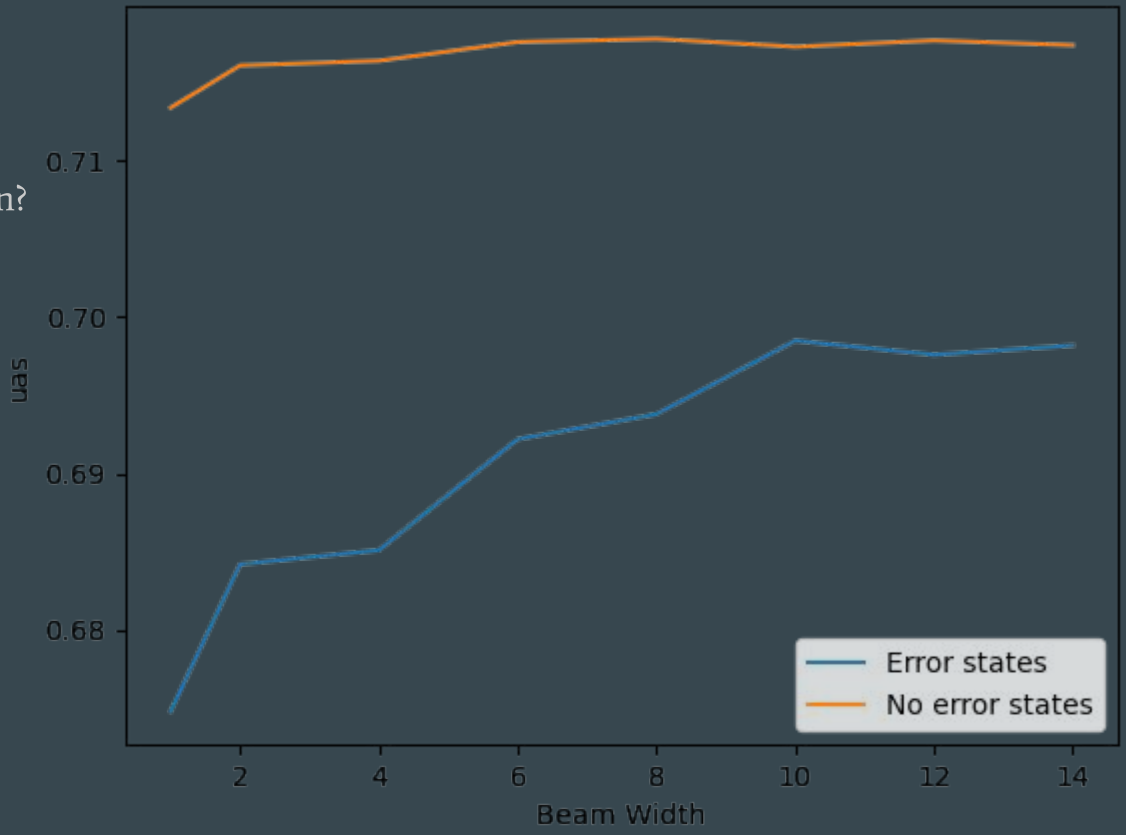
Efficient Structured Inference for Transition Based Parsing with Neural Networks and Error States

Integrated error states for transition-based parsing

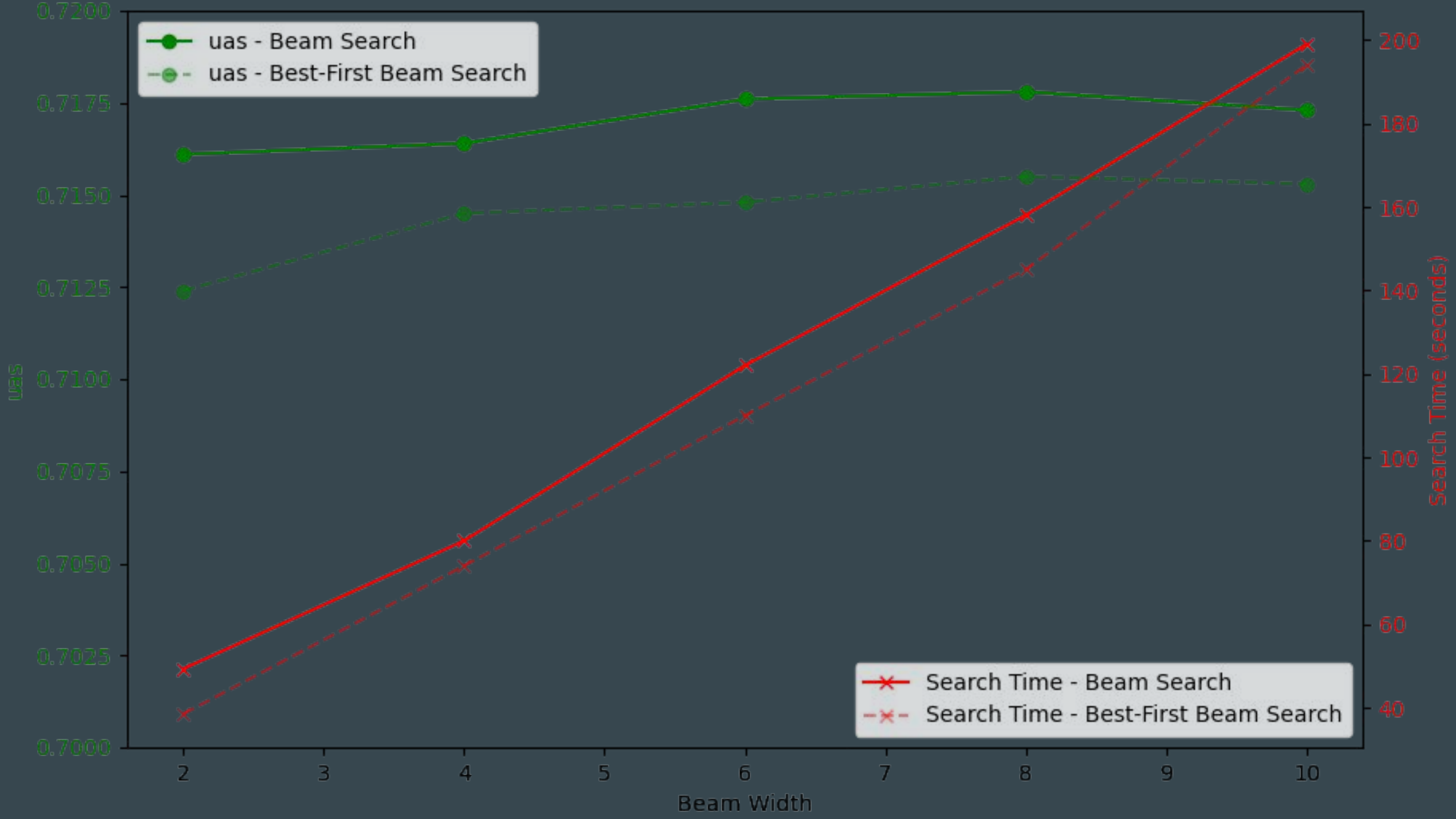
Improved accuracy on various datasets

Results

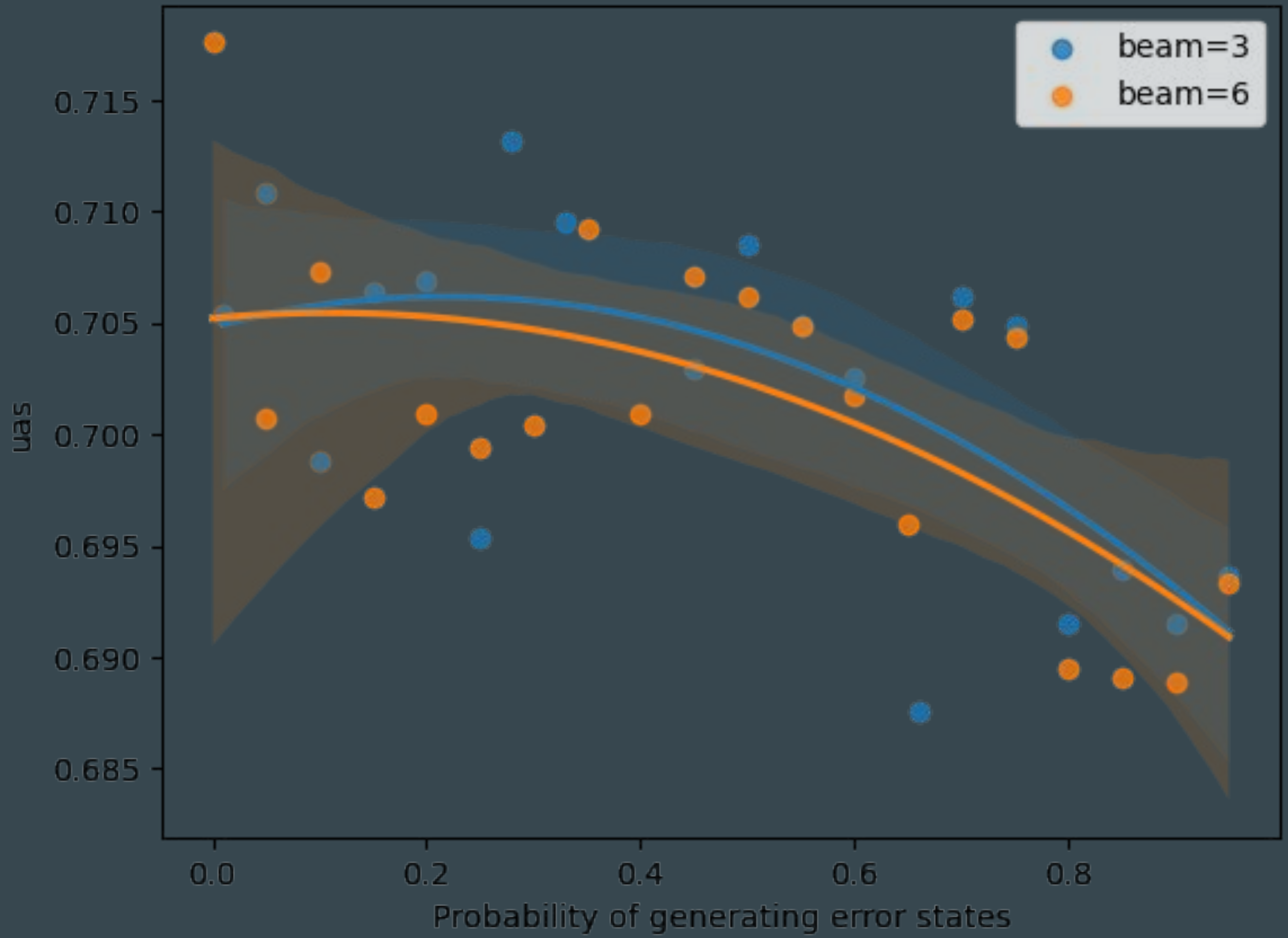
- Best beam width?
- Use Error States or not?
  - Use all Error States or a fraction?
- How well does it work for different languages?
  - Size of datasets impactful?



Best-First Beam Search



Generating less error states

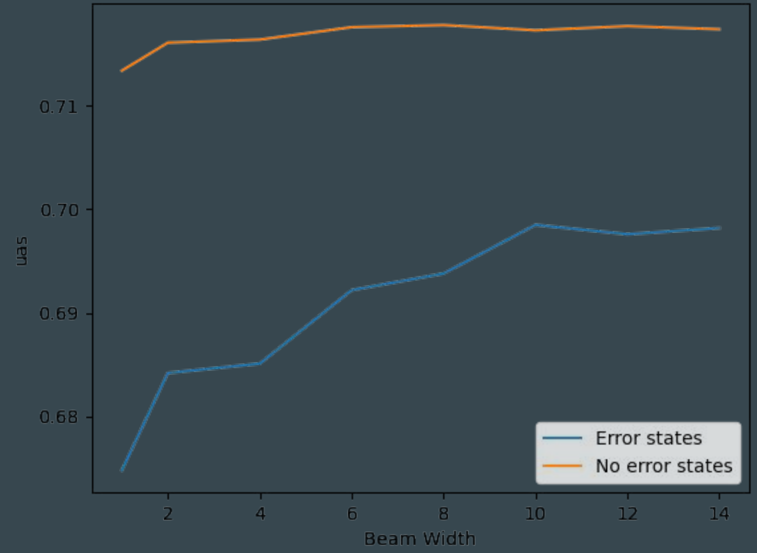


Why did it not work?

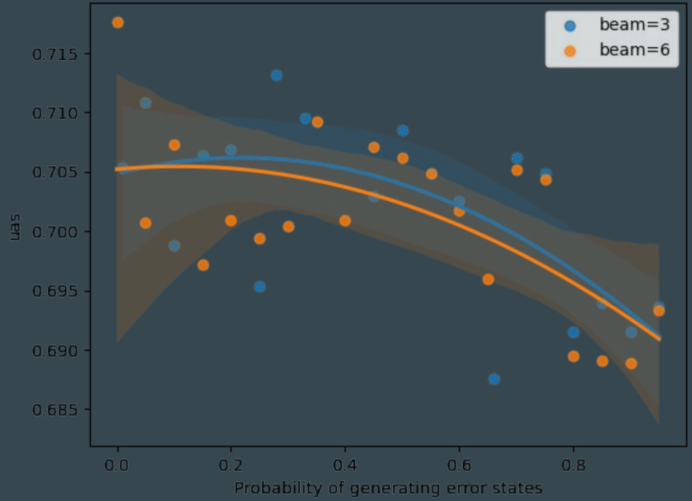
Features:

They use 14 and 25

We use 6



Dataset



Conclusions

Beam Search is better than greedy

Error states don’t necessarily improve performance

~50% of error states might be optimal