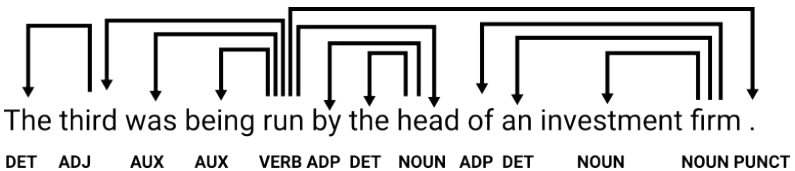
Evaluating Syntactic Parsing with Beam Search

Algot, Anna, Axel, Pontus

Background

Syntactic parsing

- Syntax tree from natural language
- Parent node of each word
- Part-of-speech tags
- Universal Dependencies treebanks



Example of syntax tree from Universal Dependencies

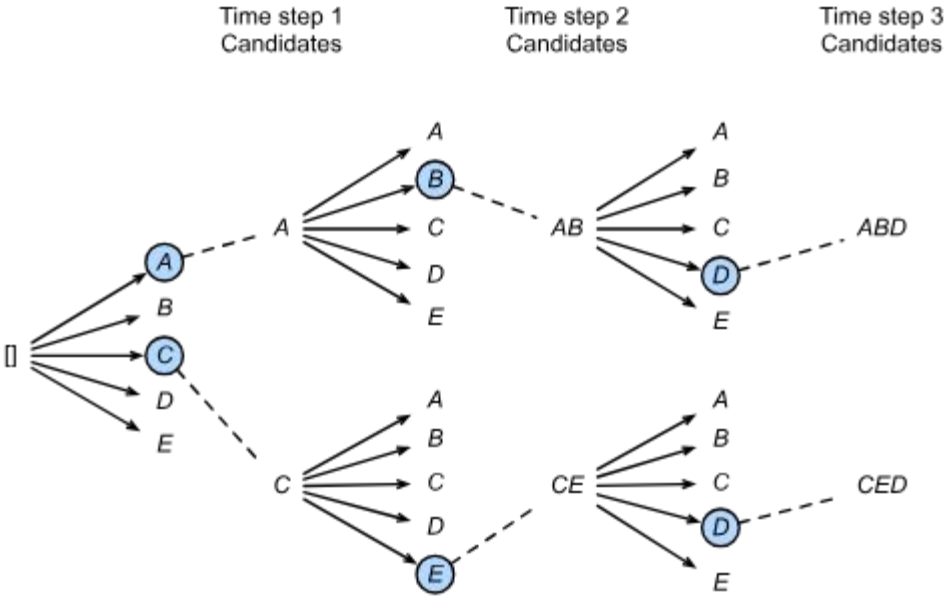
Arc standard parsing

- Introduced by Nivre (2004)
- Stack, buffer, partial tree
- Utilizes three transitions
  - Shift (SH)
  - Left-arc (LA)
  - Right-arc (RA)

Beam Search

- Explore multiple alternatives
- Beam width of size *n*
- Expands the *n* most promising nodes
- Uses these nodes to generate new alternatives
- Introduce new “transition”, error state (ER)
  - Model associates incorrect states with ER

Vaswani and Sagae (2016)



Beam Search example (d2l.ai)

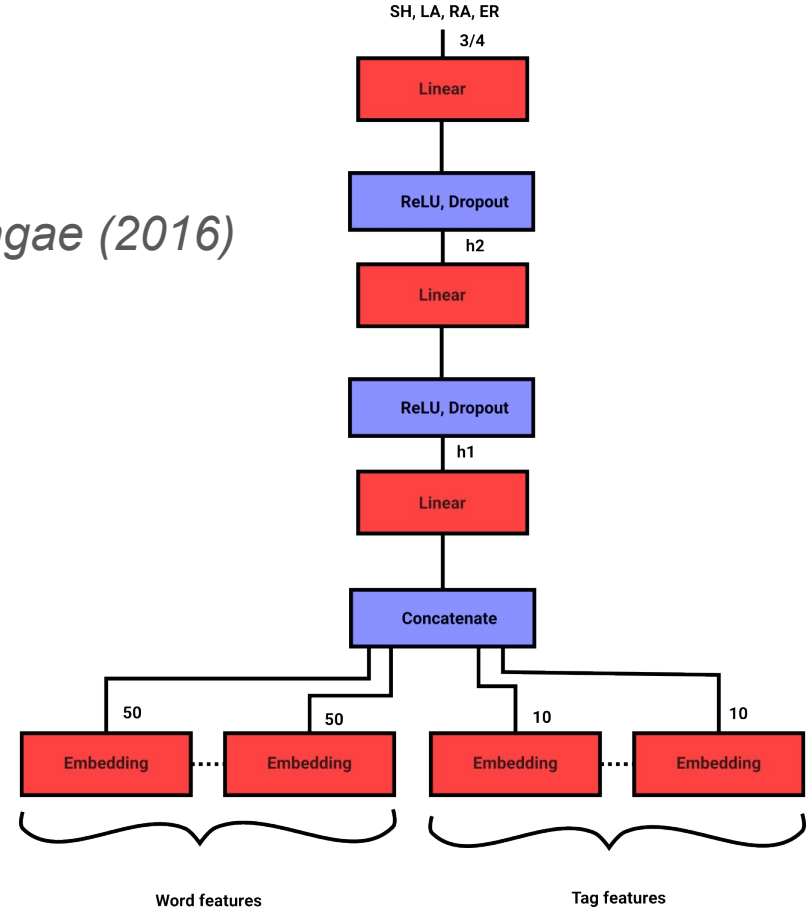
Method

Arc standard parser as baseline

Beam search during testing, *Vaswani and Sagae (2016)*

Experimentation

- Different seeds
- Error state probability
- Features and parameters
- Beam width



Results

Measured and compared unlabeled attachment score (UAS) between baseline and beam search model.

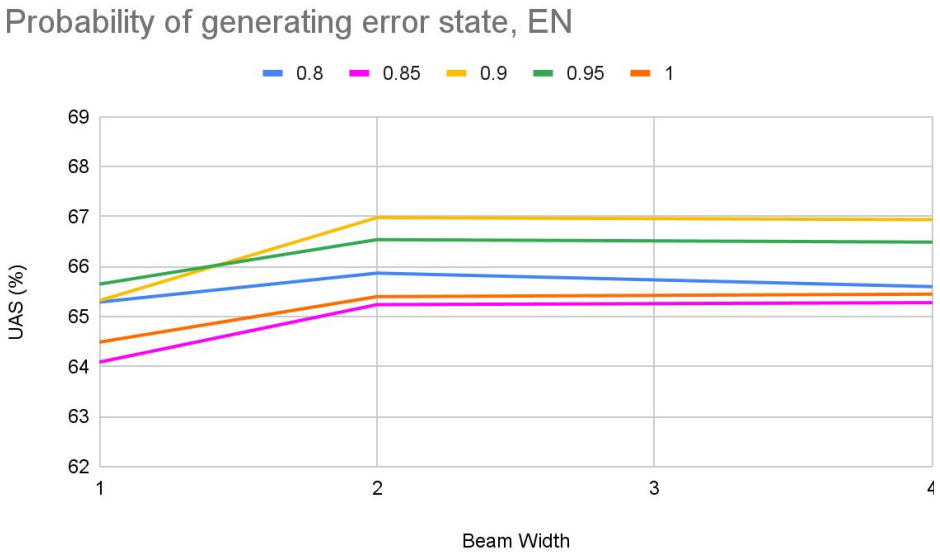
Baseline	Beam search
0.6568	0.6544

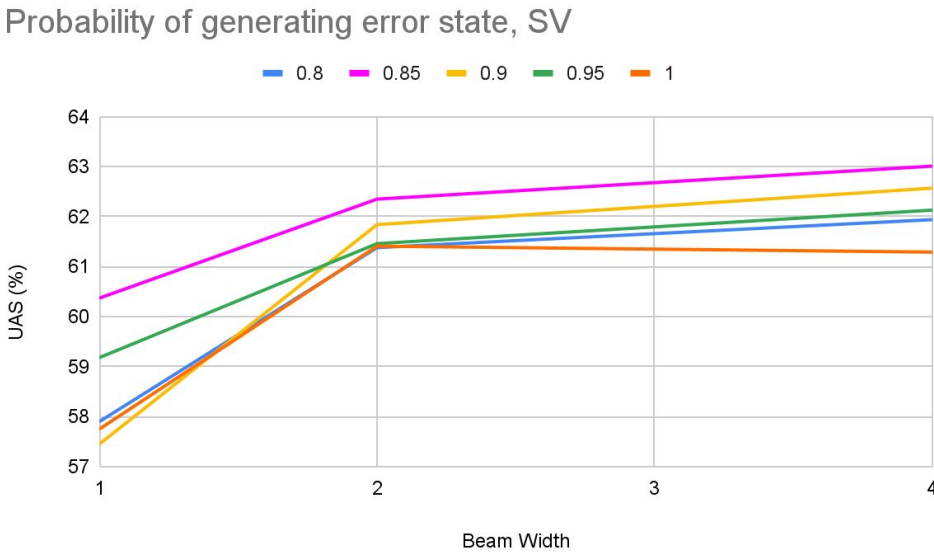
Choice of seed barely had any impact on results.

Very small difference, even with optimal seed (Picard (2023))

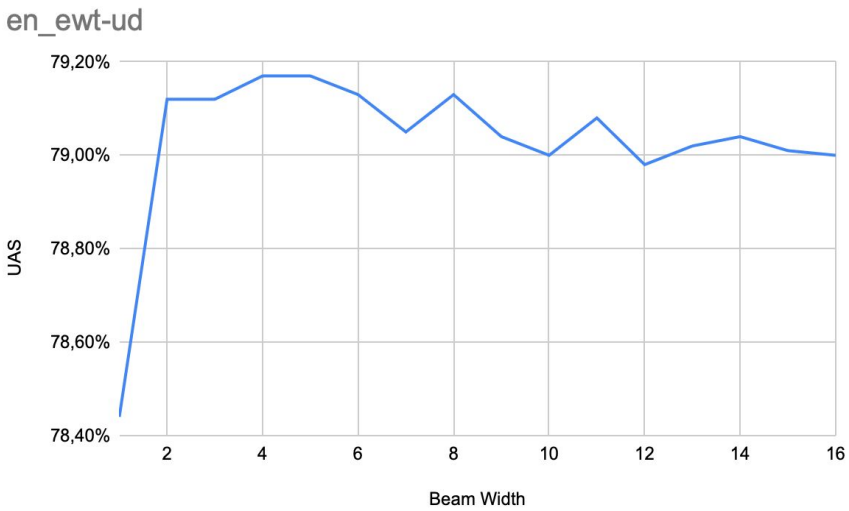
The other experiments rendered more interesting results

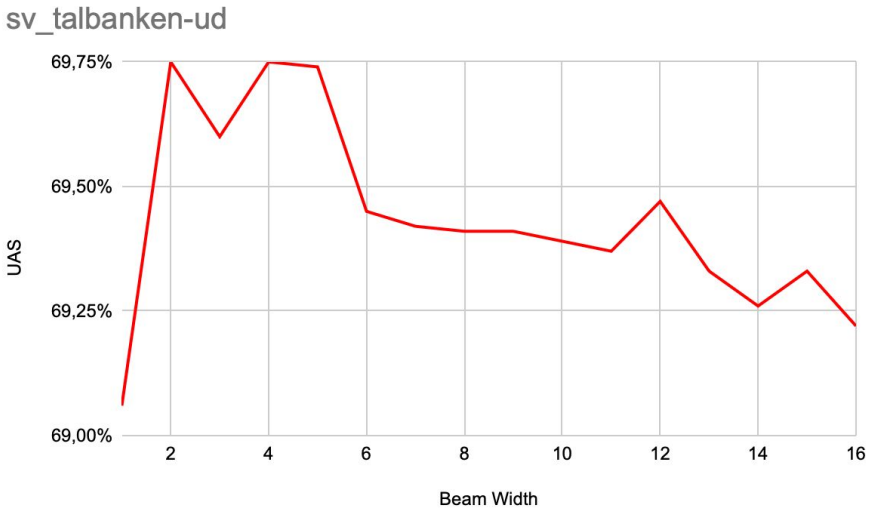
Varying Probability of Generating Error States



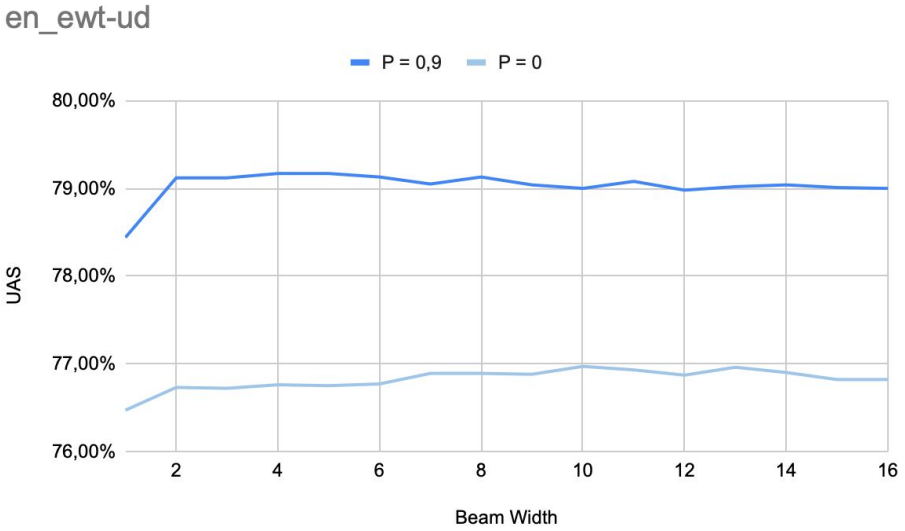


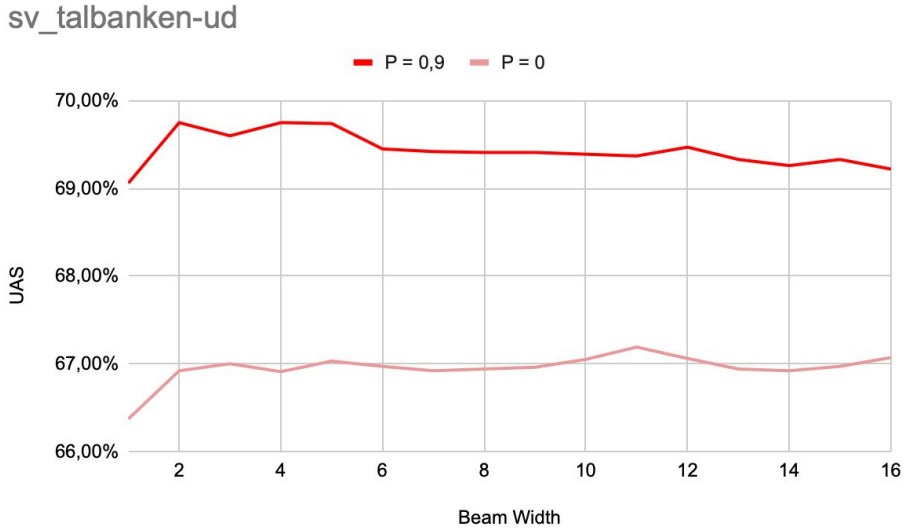
Effect of Beam Width





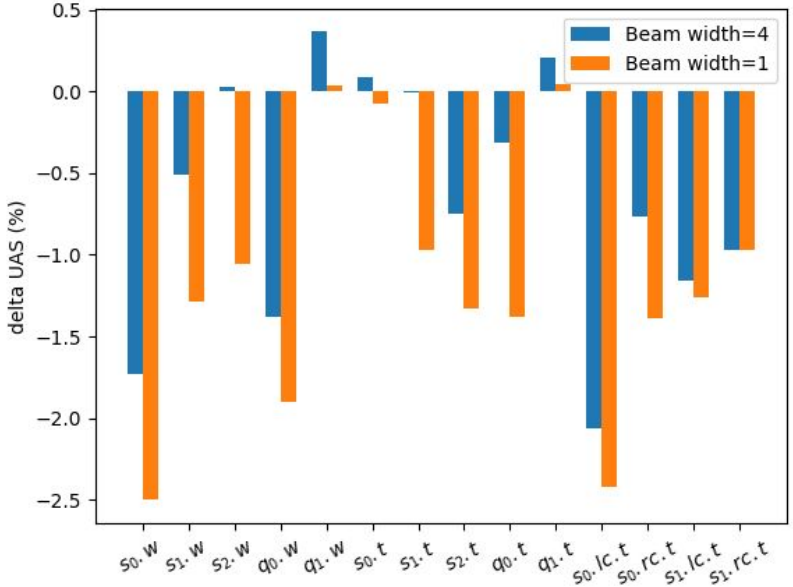
Impact of Generating Error States During Training





Effect of feature set

- Loss in UAS score from removing single features
- Choice of features is an important factor in accuracy



Conclusion

- Beam search results in a small increase in UAS which is in line with the findings of *Vaswani and Sagae (2016)*
- Features and model optimization can have a larger impact