Enhancing Dependency Parsing with Beam Search and Error State Classification

A project by:

Anton Bergman

Fabian Bergström

Danijel Grujicic

Shamil Limbasiya

Elsa Kihlberg Gawell

Introduction

- Reason for choice of project
- Syntactic Parsing

The Baseline

- Baseline project
  - Tagger
  - Parser

Part-of-speech tagging

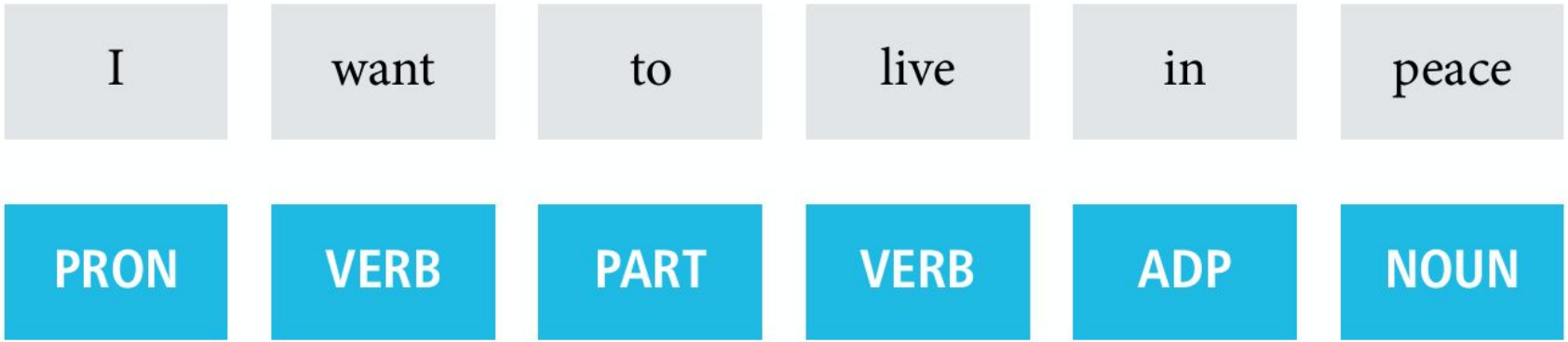
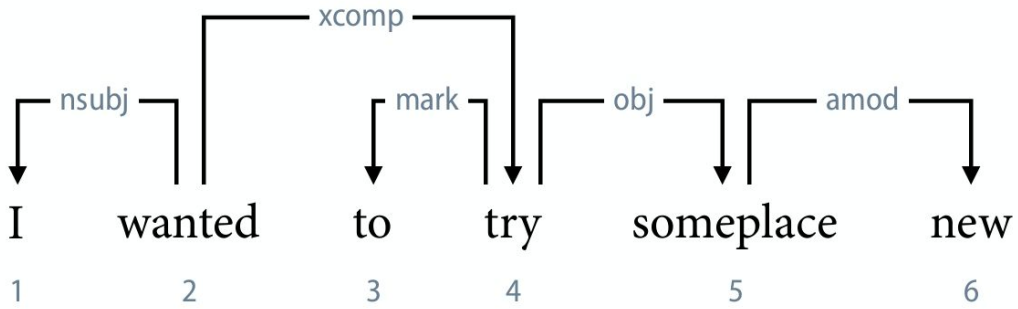


Figure: NLP-TDDE09, Liu, Lecture 4 by M. Kuhlmann

Representation of dependency trees



word position	1	2	3	4	5	6
head position	2	0	4	2	4	5
dependency relation	nsubj	root	mark	xcomp	obj	amod

Figure: NLP-TDDE09, Liu, Lecture 4 by M. Kuhlmann

The Baseline

- Evaluation
    - Unlabeled Attachment Score - UAS

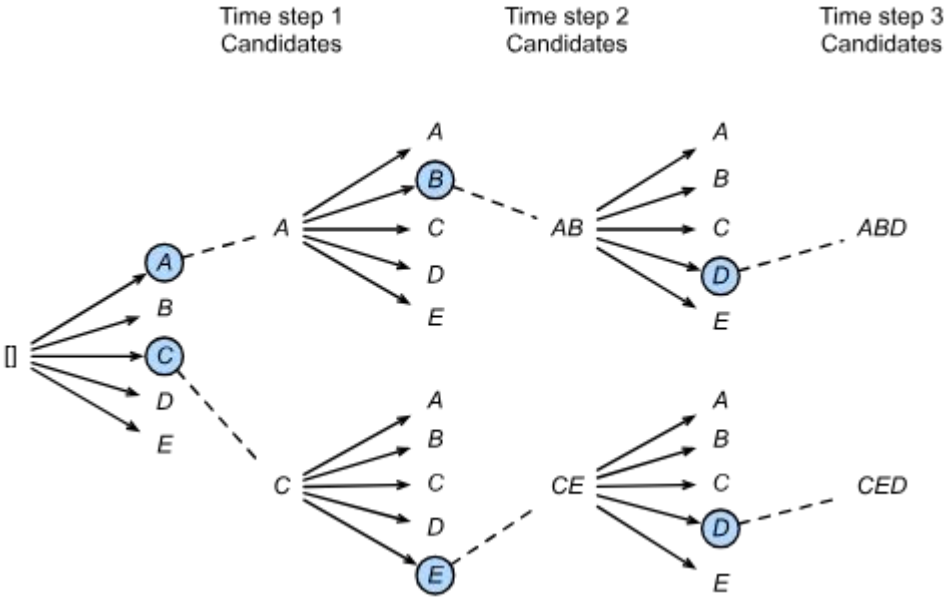
Baseline system score

Language	Dataset	Tagging Accuracy	UAS
English	EWT	0.8788	0.6681
Swedish	Talbanken	0.9040	0.6236

Beam-search

- beam size *k*
- continue k paths with highest probability
- Integrating Beam-search on top of Baseline

k = 2:



src: Dive into Deep Learning (https://d2l.ai/index.html)

Beam-search implementation

- Vaswani & Sagae, 2016: https://aclanthology.org/Q16-1014.pdf
- Zhang & Clark, 2008: https://aclanthology.org/D08-1059.pdf
- Fixed Window Parser With Beam
  - inherits from Fixed Window Parser in baseline
  - overwrites predict function to return predicted heads based on beam-search

What are Error States?

- Errors occur when the parsing model makes an incorrect decision during the transition process.
- This can happen due to various reasons:
  - Insufficient information for the model to make the right choice.
  - Model limitations stemming from training data or internal biases.
- Dealing with these errors using error states in training of the parser

Dealing with Error States

Backtracking:

- This approach involves revisiting previous decisions (transitions) and exploring alternative paths.
- Buckman et al. (2016) explored this method using confidence estimates to guide the search for the optimal parse, achieving accuracy comparable to beam search.
- Ref: https://aclanthology.org/D16-1254.pdf)

Dealing with Error States

Using complex models:

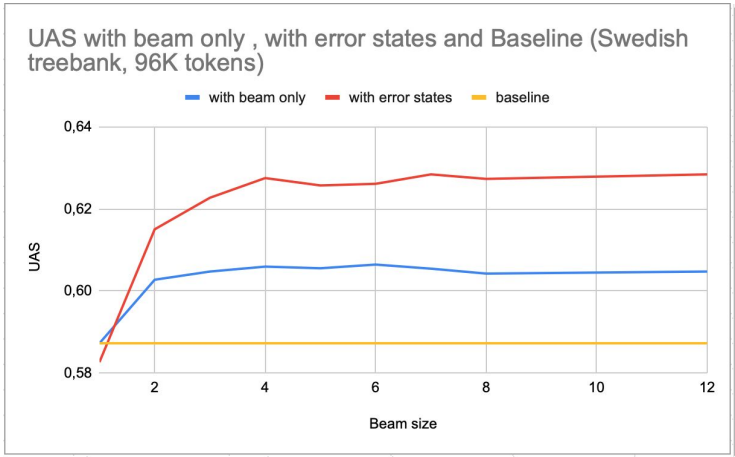
- A more complex model may better capture the intricacies of human language and avoid making errors in the first place.
- Dyer et al. (2015) explored this concept with a Stack-LSTM model, a specialized Recurrent Neural Network designed for transition-based parsing.
- (Ref: https://arxiv.org/abs/1505.08075)

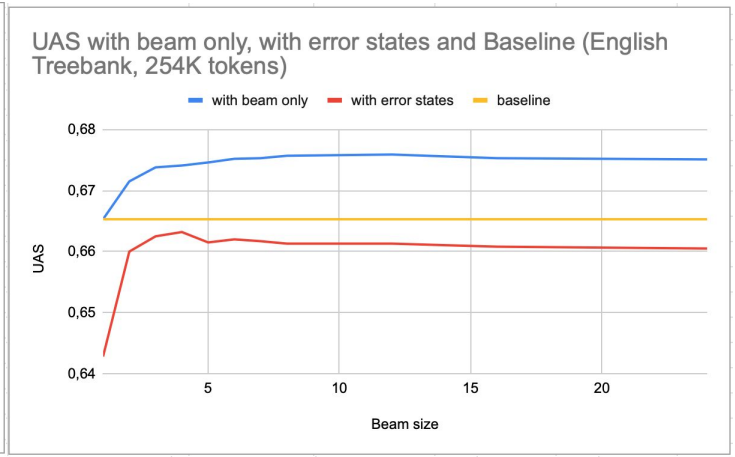
Dealing with Error States

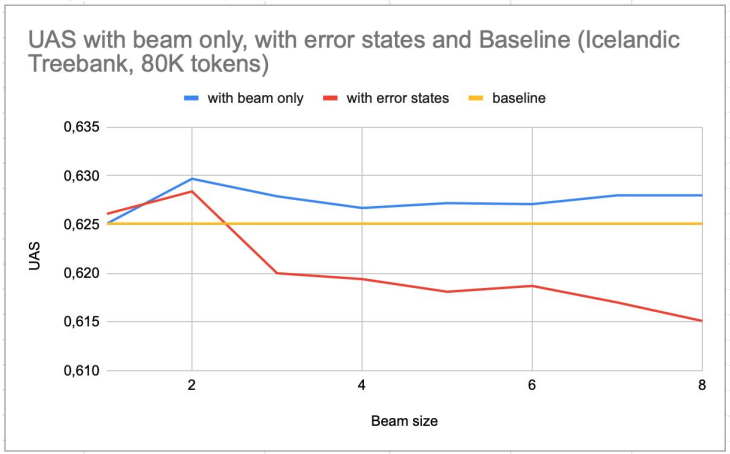
Beam search (our approach):**

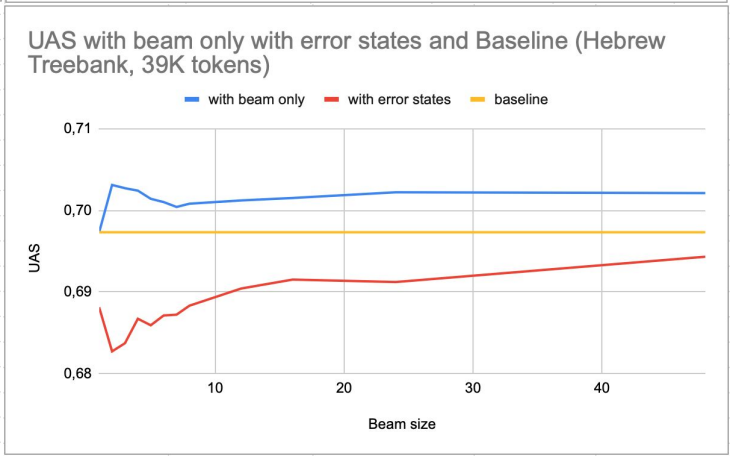
- This method involves keeping track of multiple potential parse trees simultaneously.
- The model discards less promising ones and focuses on the most likely candidates.
- This is the approach we implemented, based on Vaswani and Sagae (2016).
- Using error states to train the network for identifying parsing pitfalls and utilizes beam search to explore promising options while avoiding dead ends.
- (Ref: https://aclanthology.org/Q16-1014/)

Experiments, language treebanks, beam size

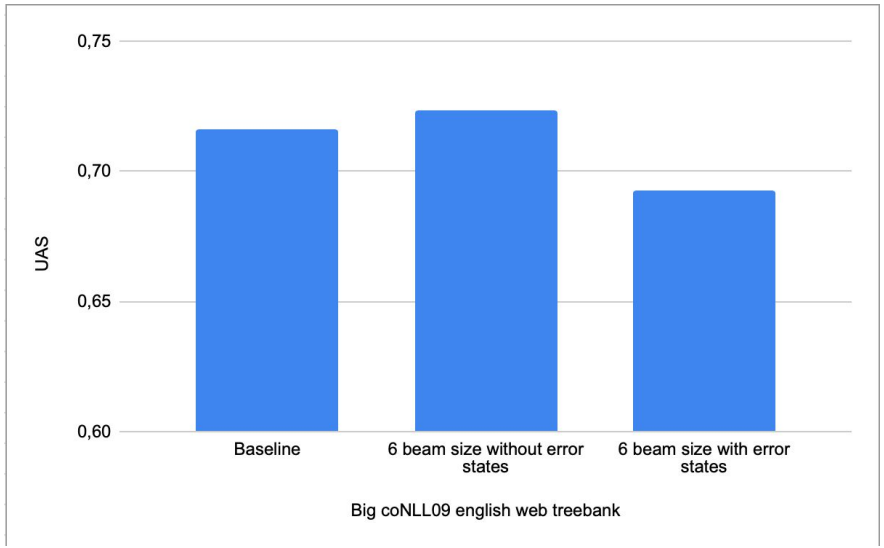


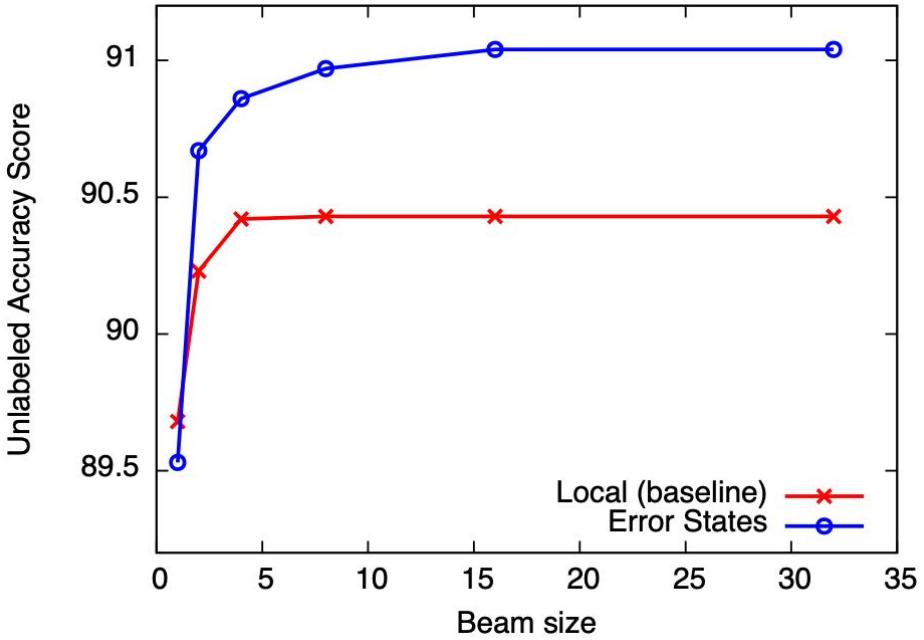






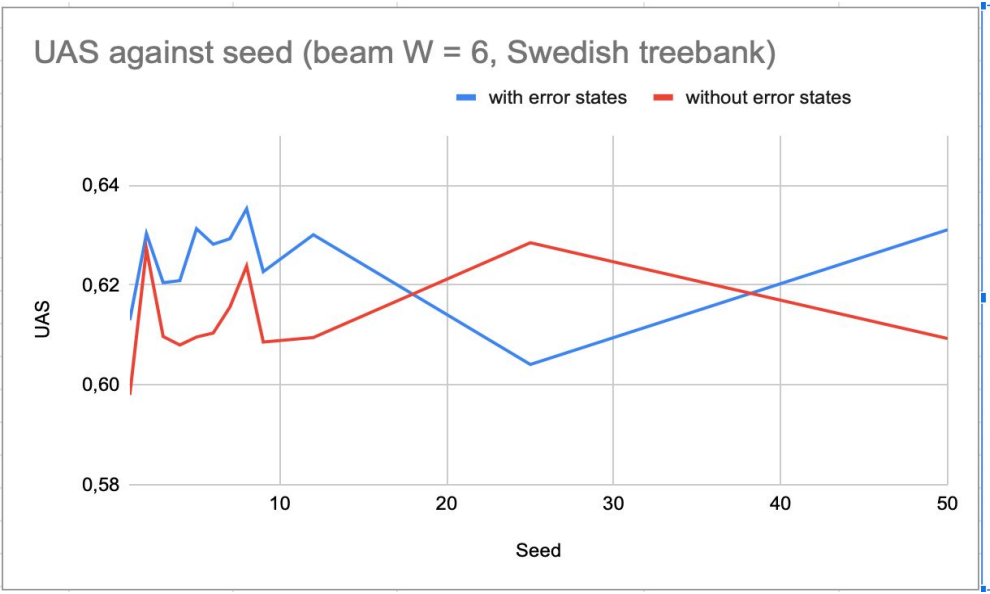
Our result vs Vaswani and Sagae

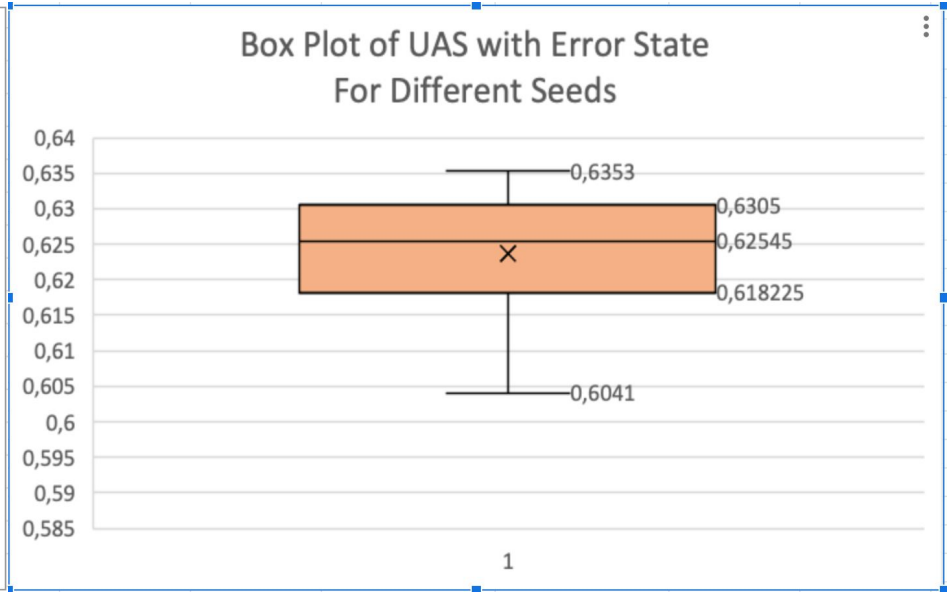




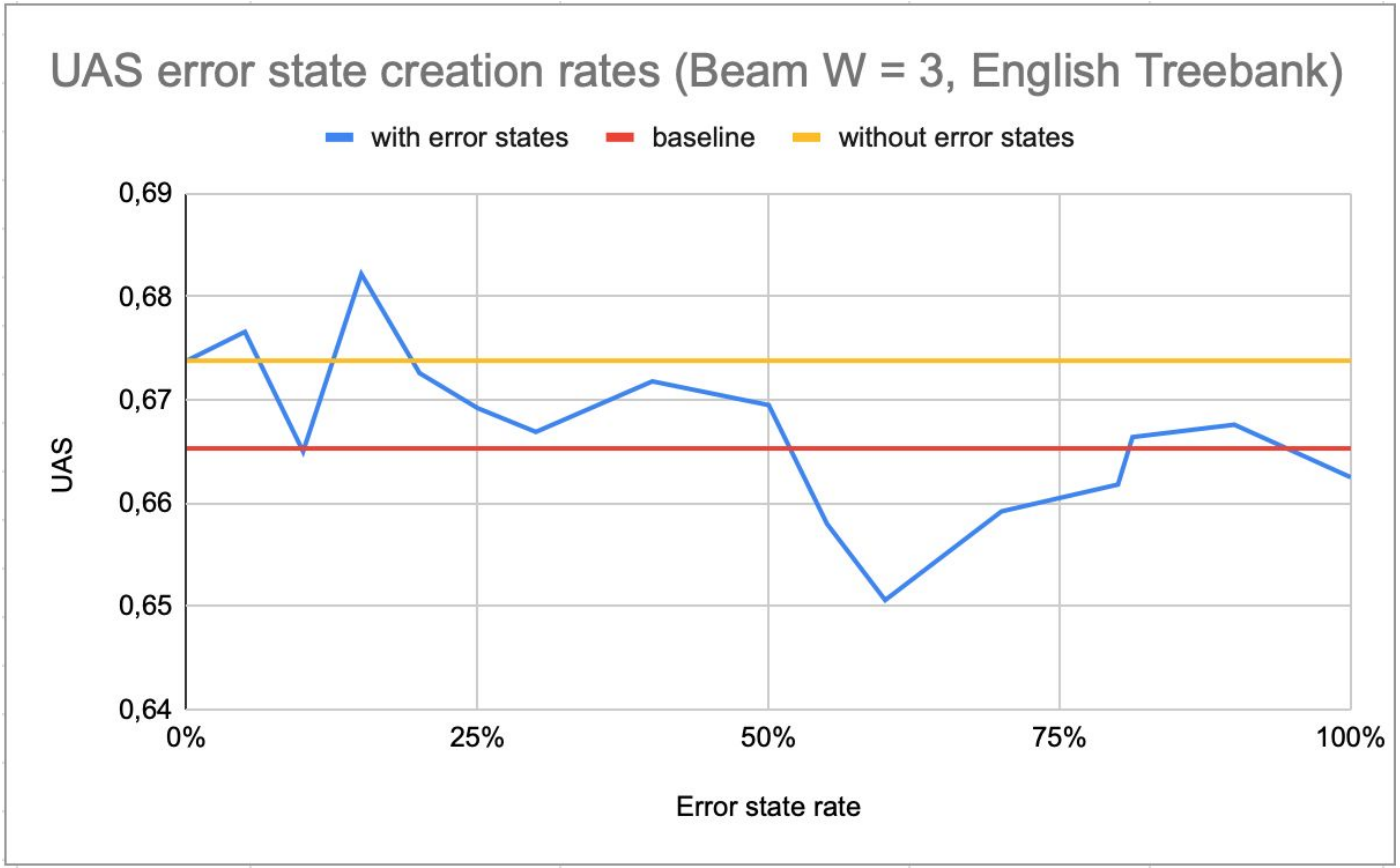
Vaswani and Sagae (penn treebank)

Seeds

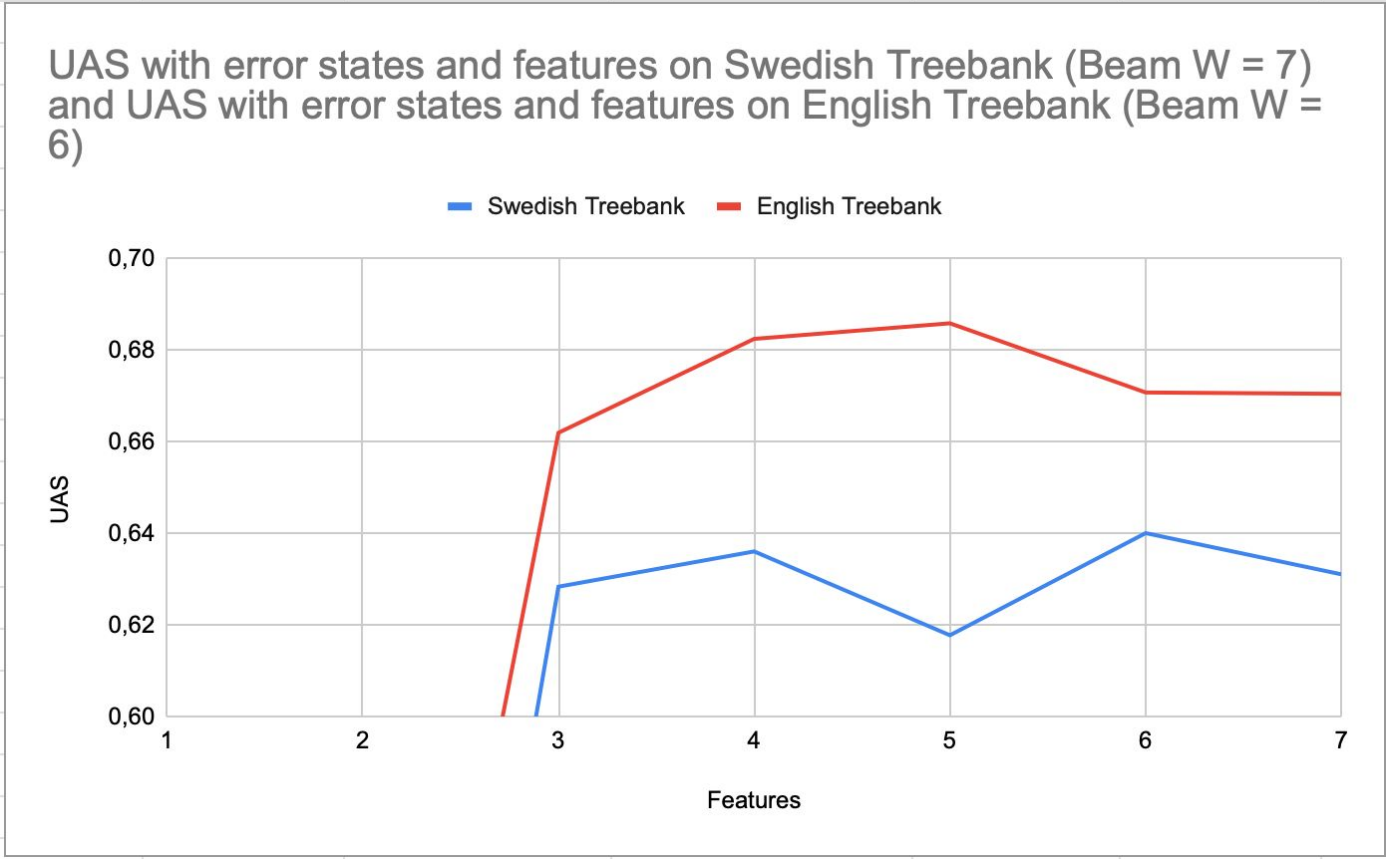

UAS against seed (beam W = 6, Swedish treebank)
with error states
without error states
0,64
0,62
0,60
0,58
UAS
10
20
30
40
50
Seed


Box Plot of UAS with Error State
For Different Seeds
0,64
0,635
0,63
0,625
0,62
0,615
0,61
0,605
0,6
0,595
0,59
0,585
0,6353
0,6305
0,62545
0,618225
0,6041
1

Error state rate



Number of features



Conclusion & Analysis

- The impact of error states
- Beam-search integration
- Differences between language tree banks
- Beam size optimum

Thank you for your attention!