TDDE09 Project Presentation Group 03

Loïs Bilat
Goutam Bhat
Jonas Rotter
Paul Thiele

LINKÖPING UNIVERSITY

2018-03-06

Outline

- Baseline System and Feature engineering
- Perceptron based
  - Non-projective trees
  - Beam search
- Neural network based
  - POS tagger
  - Syntactic parser

Baseline system

- 64.60% (65.18%) accuracy on English dataset
- 59.60% (62.13%) accuracy on Swedish dataset
- 74.88% (71.46%) accuracy on French dataset

After feature engineering

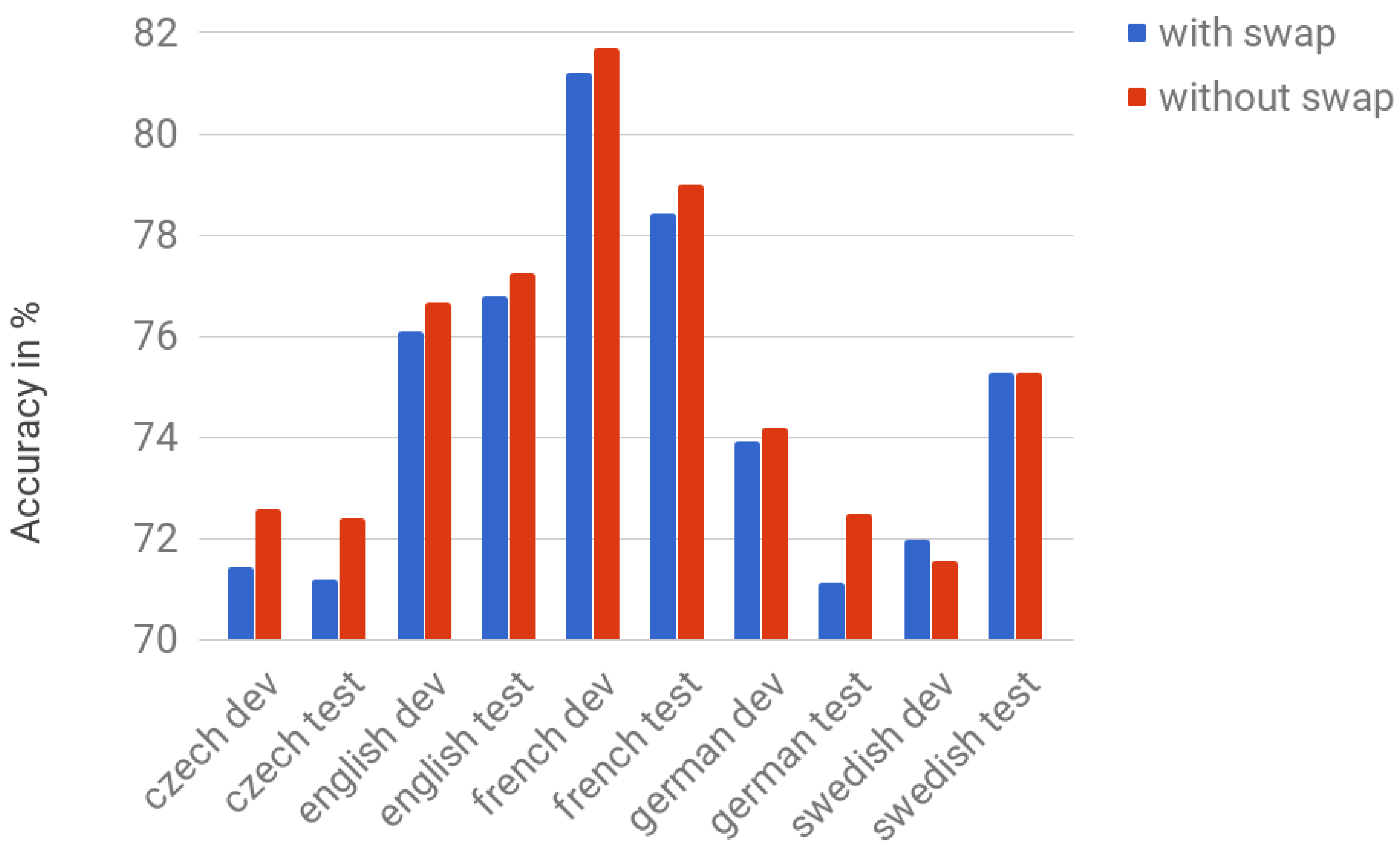
- 76.11% (76.79%) accuracy on English dataset
- 71.98% (75.28%) accuracy on Swedish dataset
- 81.22% (78.43%) accuracy on French dataset

Non-projective trees

- Non-projective trees exist in reality
- Possible improvement in accuracy
- Introduce "swap"-operation

- Research article:
  Non-Projective Dependency Parsing in Expected Linear Time

Non-projective trees - Results



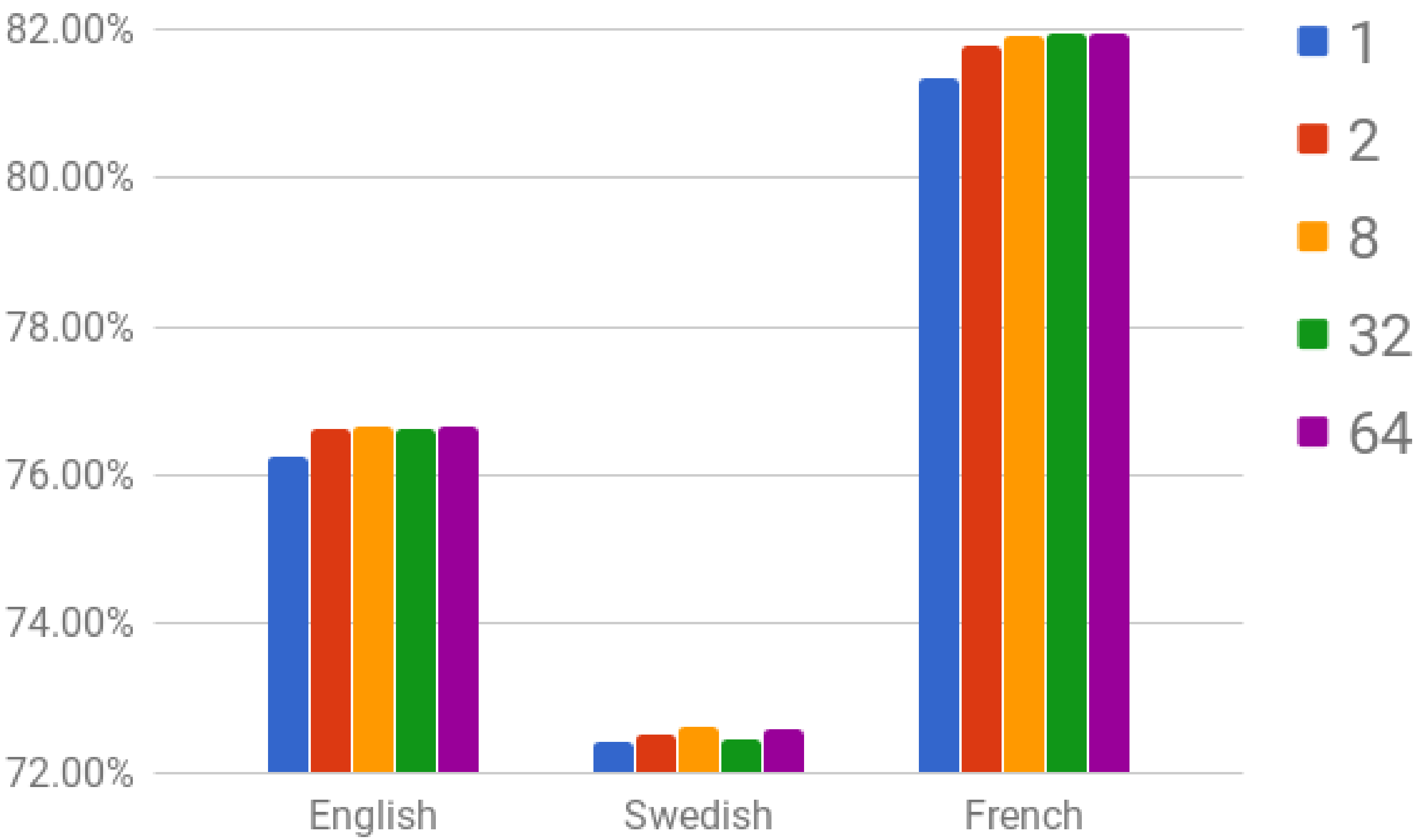
LINKÖPING UNIVERSITY

Beam Search

- During the transition-based parsing
- At each step, apply all possible moves, not only the predicted one
- Always consider multiple partial dependency trees at the same time
- Reduces error propagation
- Similar to a Breadth-first search
- Can be implemented for both training and parsing
- Research article: **A Tale of Two Parsers**

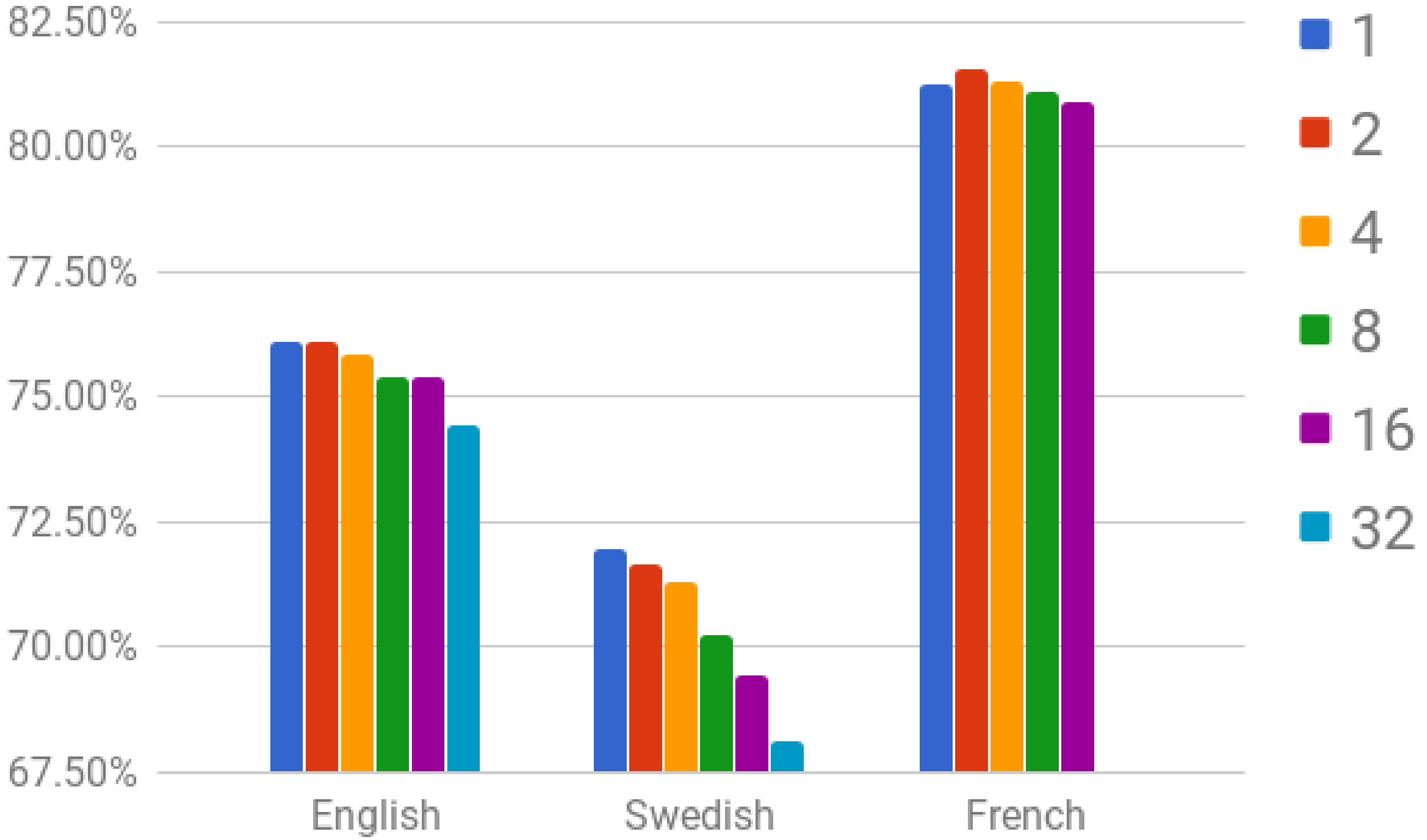
Beam Search - Results

- Beam search for parsing only



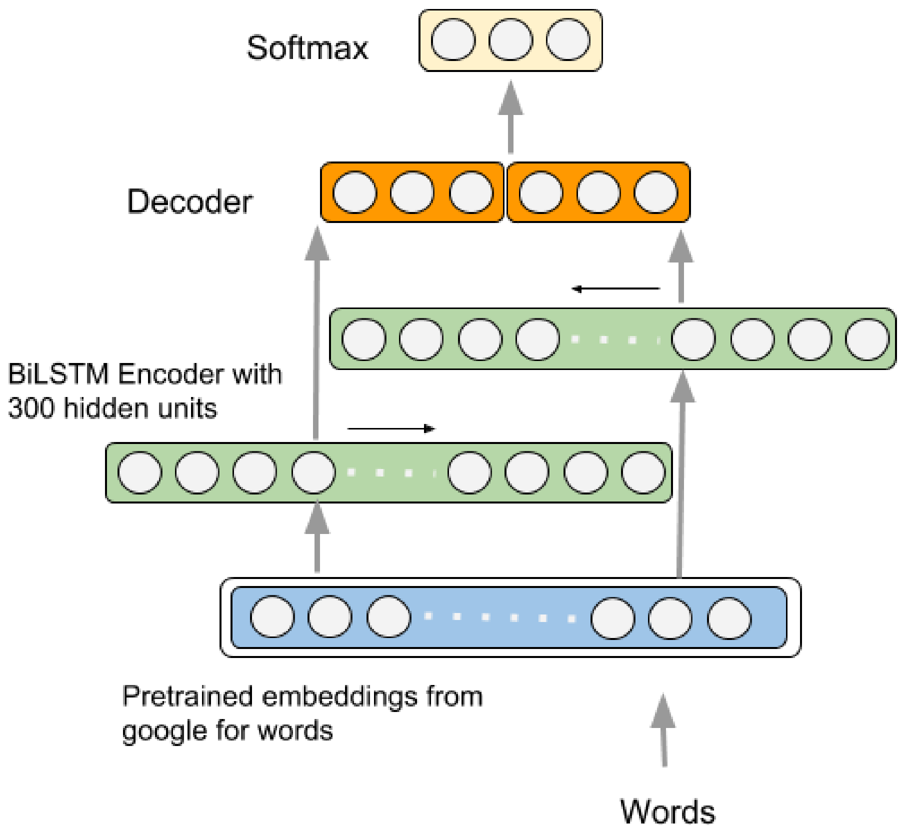
Beam Search - Results

- Beam search for training and parsing



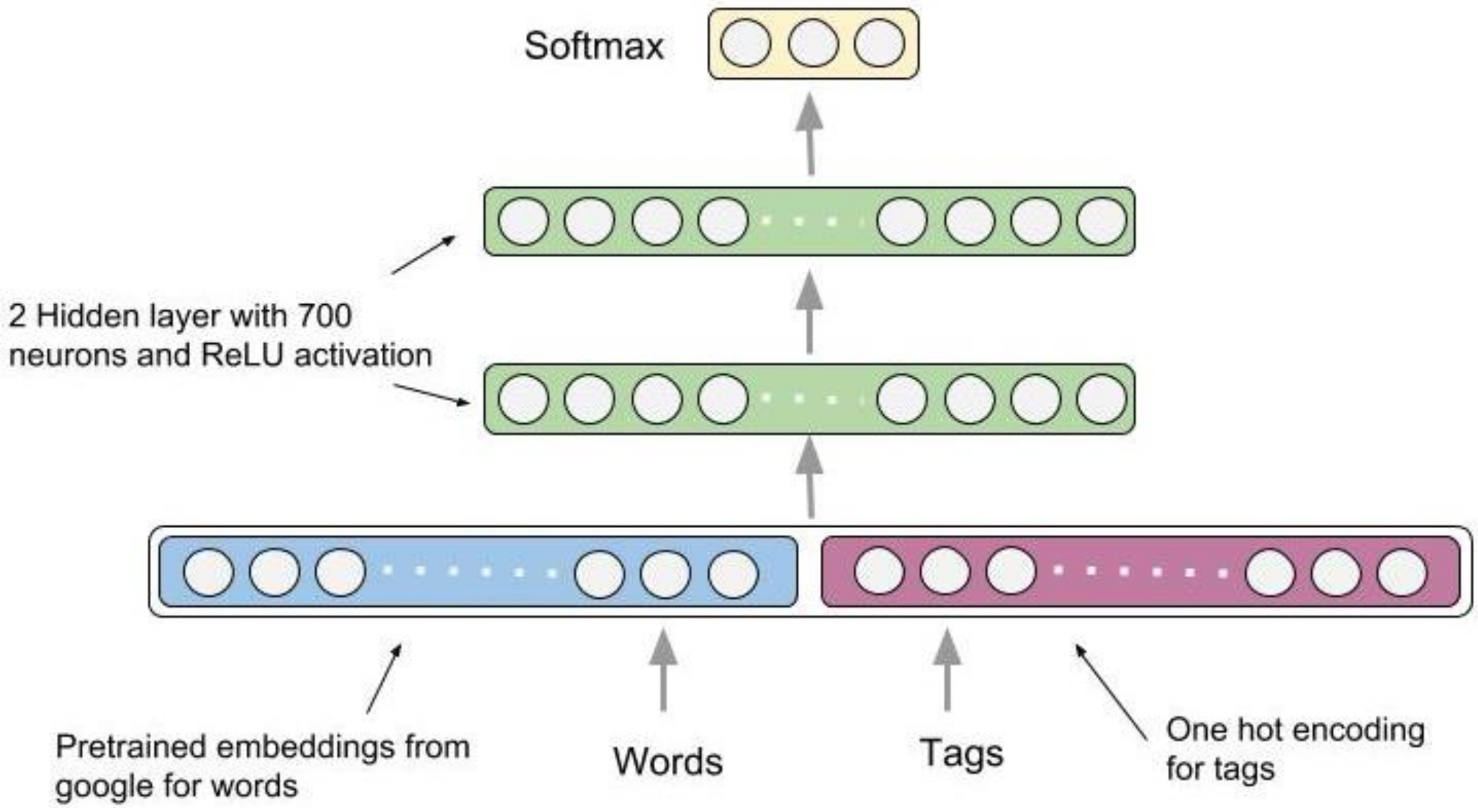
NN - POS Tagger

- Based on: POS Tagging with Bidirectional LSTM RNN https://arxiv.org/pdf/1510.06168.pdf



NN - Dependency Parser

- Based on A [Fast and Accurate Dependency Parser using Neural Networks]() by Chen and Manning



NN - Results

Score on the test data

	Tagging Accuracy	UAS	Exact matches
LSTM	88.97	79.81	47.47
Neural Net	92.91	79.77	48.94
Perceptron	92.94	80.45	48.67

- Observation: Word embeddings help generalize to unseen words.
  - ~2 % improvement when using pretrained word embeddings for unseen word, as opposed to using a random embedding

- Note: Due to random initialization of the weights, the results can vary between different runs. The values provided are average over 3 runs

Conclusion and Outlook

- Reimplemented 4 topics
  - features ,"swap", beam-search, neural networks
- Largest improvement by feature engineering (~10%)
- Less improvement compared to the papers
- Many more possibilities:
  - Pretrain word embeddings
  - Larger training data sets
  - Combine approaches:
    - e.g. Beam search with Neural Network
    - Conditional Random Field to predict a Viterbi-sequence