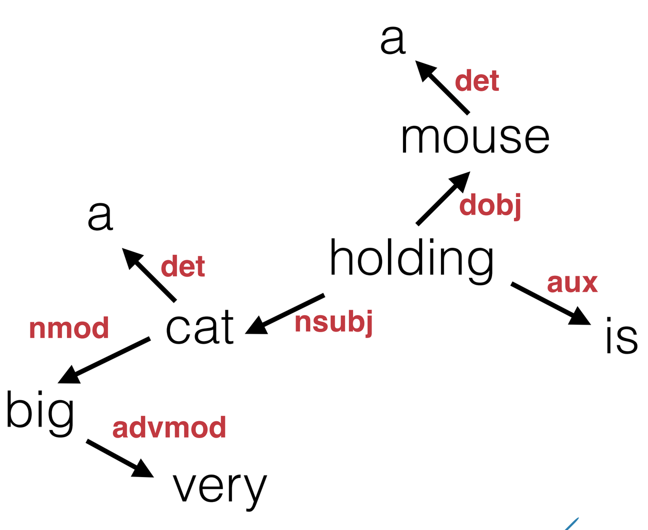
Incremental generative model of sentence linearization and word order variation

Kristina Gulordava — University of Geneva

Frank Keller — University of Edinburgh

Language production and sentence linearization

underlying representation → utterance



"a very big cat is holding a mouse"

Task: unordered dependency tree → order of words

Research goal: a cognitive model

How can sentence linearization be performed word-by-word?

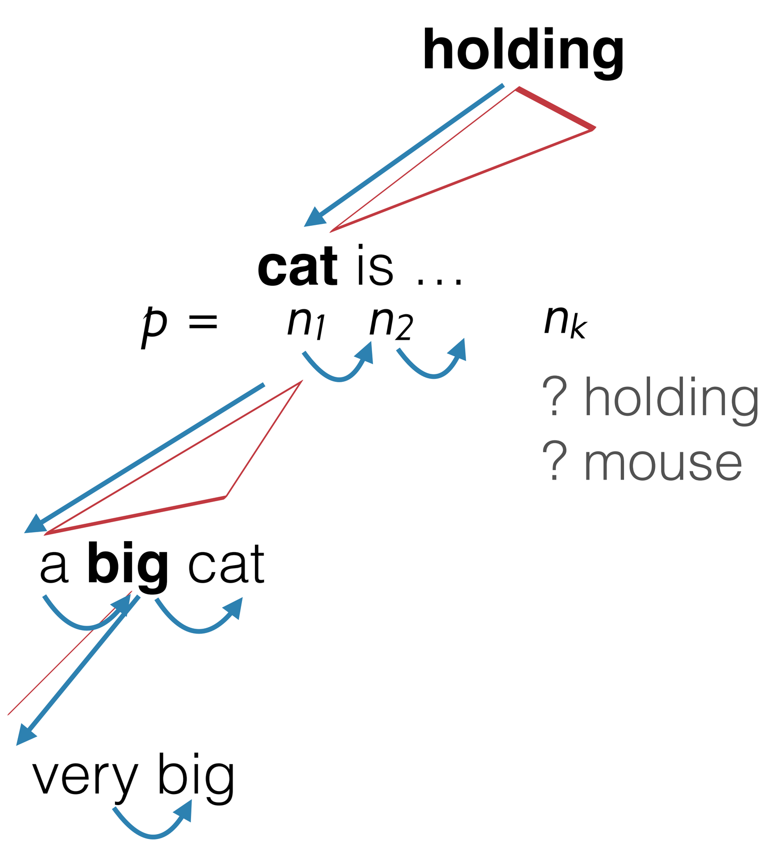
Incrementality

- Speakers don't plan the whole utterance in advance
- What is a plausible degree of incrementality? word-by-word, chunk-by-chunk

Probabilistic nature

- Speakers have access to a probabilistic grammar (e.g. for processing)
- How are these probabilities used in generation?

Linearization process



Recursive procedure

- tree is traversed **top-down**
- each set of *head + immediate children* is ordered **independently**

Greedy choice of the next node in each set results in word-by-word linearization

Score function defines which node is chosen given previously output nodes *p* and the remaning nodes in the set

Probabilistic score function

$$score(p) = \prod_{n_i \in p} score(n_i, p) \cdot \prod_{n_j \notin p} score'(n_j, p)$$

***generation score** for nodes in p*

$P(n_i | n_1 \dots n_{i-1}) \cdot P(left | n_i, head)$

***ngram** probability* — ***direction** probability*

***future score** for remaning nodes*

if *head* is in the output nodes *p*: $score'(n_j, p) = P(right | n_j, head)$

else: $max \begin{array}{l} P(left | n_j, head) \\ P(right | n_j, head) \end{array}$

We estimate the **unlexicalized** probabilities from a treebank:

- conditioned on *dependency label*, *part-of-speech tag* (no token information)
- ngram probabilities are estimated as *trigrams*; no smoothing

Re-ranking with size features

Modelling word order variation cases

A cat is staring [at a poor little mouse] [with a hungry look]
A cat is staring [with a hungry look] [at a poor little mouse]

- two alternative grammatical orders with the same semantics, i.e. unordered dependency trees
- some relevant features: **sizes of the phrases**
- choice between two options *(..., staring, mouse) vs (..., staring, look)* can be modelled as a discriminative re-ranking at each linearization step

best two hypotheses: n_k^1, n_k^2, n_k^3, ... → *features*: $score(n_k^1)$, $pos(n_k^1)$, $dep\_label(n_k^1)$, **$size(n_k^1)$**; $score(n_k^2)$, $pos(n_k^2)$, $dep\_label(n_k^2)$, **$size(n_k^2)$** → binary classifier → n_k

Data and set-up

Four UD treebanks: English, Italian, Persian, Russian (development sets)

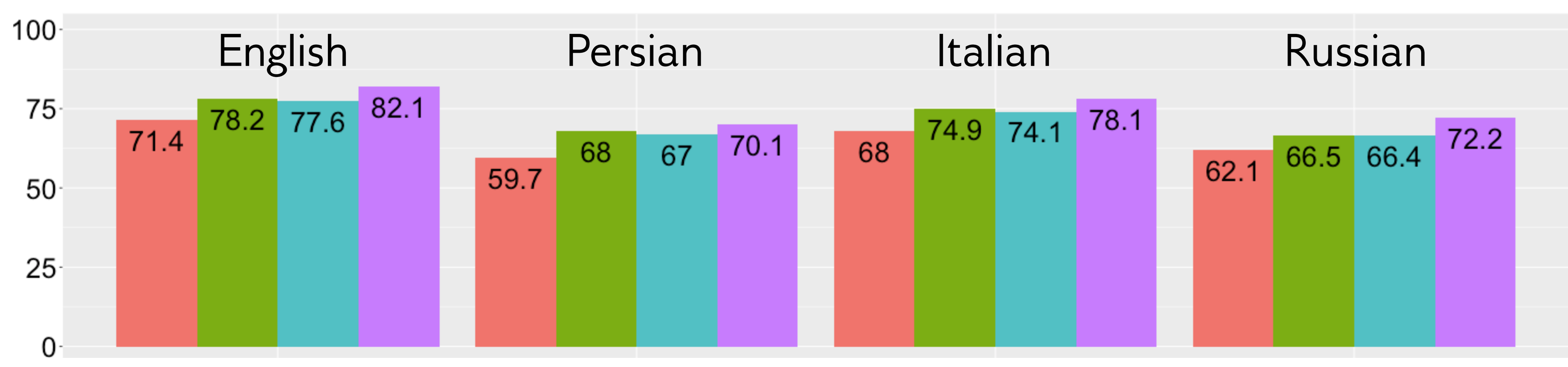
Pre-processing: only sentences without punctuation

Point of comparison: **ZGen** (Liu et al, 2015) - state-of-the-art transition-based linearization system; lexicalized, **uses large beam (64)**

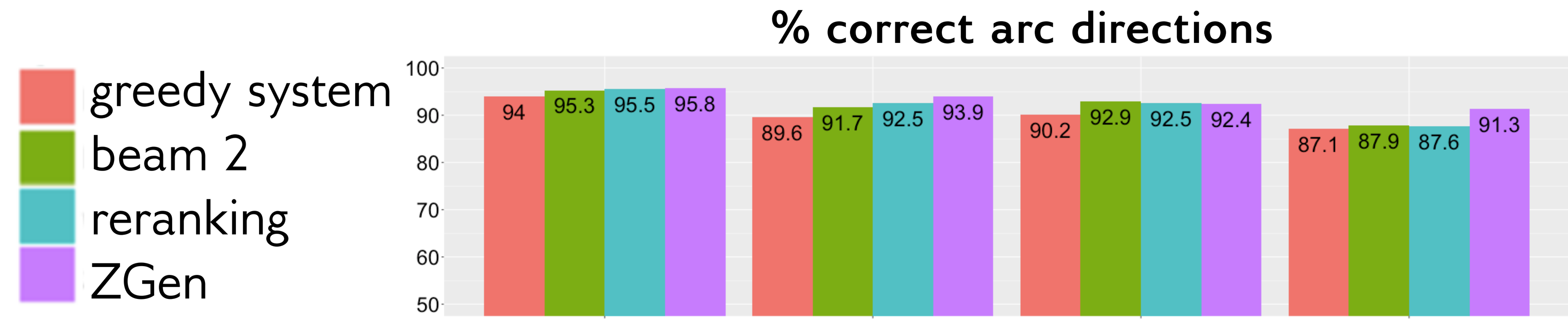
Measures: **BLEU** and **% of arcs having correct direction**

Results

BLEU scores



% correct arc directions



- Purely incremental system has lowest performance but it's **only ~10 BLEU points lower** despite its very simple greedy architecture
- Keeping **two hypotheses** instead of one at each linearization step (beam 2) improves the results by up to **8 BLEU** points

- Reranking improves significantly over the greedy system, reaching almost the performance of the system with beam 2
  - discriminative information in terms of two best nodes is crucial
  - confirms that size features play role in choosing better word orders

Conclusion

We can reach competitive performance using a cognitively plausible architecture with greedy search, probabilistic score function and unlexicalized features