# Delimiting Morphosyntactic Search Space with Source-Side Reordering Models

Joachim Daiber, Khalil Sima'an

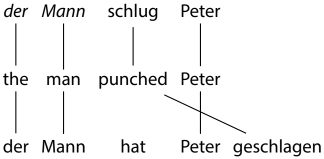
*Institute for Logic, Language and Computation*
*University of Amsterdam*

EXPERT

# Motivation

- Current MT models work well if languages are structurally similar
- Difficulties with morphologically rich languages:
  - freer word order
  - more productive morphological processes
  - agreement over long distances

# Motivation




**"Germans like to buy holiday homes in Florida"**

- Deutsche kaufen sich meistens in Florida eine Ferienwohnung
- Deutsche kaufen sich in Florida meistens eine Ferienwohnung
- In Florida kaufen sich meistens Deutsche eine Ferienwohnung
- In Florida kaufen sich Deutsche meistens eine Ferienwohnung
- Meistens kaufen sich Deutsche in Florida eine Ferienwohnung
- ...

From: *Frankurter Allgemeine Zeitung (August 31, 2015)*

# Motivation

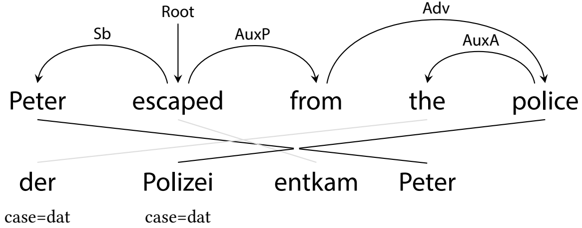



**“Germans like to buy holiday homes in Florida”**

- Deutsche kaufen sich meistens in Florida eine Ferienwohnung
- Deutsche kaufen sich in Florida meistens eine Ferienwohnung
- In Florida kaufen sich meistens Deutsche eine Ferienwohnung
- In Florida kaufen sich Deutsche meistens eine Ferienwohnung
- Meistens kaufen sich Deutsche in Florida eine Ferienwohnung
- ...

From: *Frankurter Allgemeine Zeitung (August 31, 2015)*

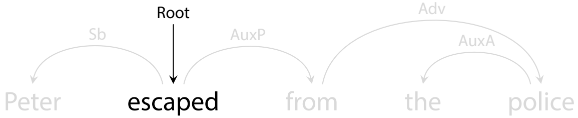
# Preordering source trees




- Source dependency trees are good fit for preordering:
  - Lerner and Petrov (2013) present two classifier-based dep. tree preordering models
  - Jehl et al. (2014) and de Gispert et al. (2015) preorder dep. trees via branch-and-bound search

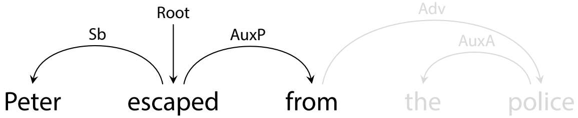
# Preordering source trees

- Lerner and Petrov (2013) preorder trees starting at the root
- Order all children (model 1) or left and right children (model 2)



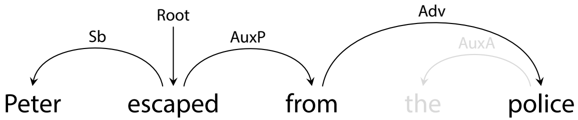
# Preordering source trees

- Lerner and Petrov (2013) preorder trees starting at the root
- Order all children (model 1) or left and right children (model 2)



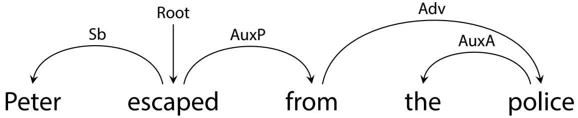
# Preordering source trees

- Lerner and Petrov (2013) preorder trees starting at the root
- Order all children (model 1) or left and right children (model 2)



# Preordering source trees

- Lerner and Petrov (2013) preorder trees starting at the root
- Order all children (model 1) or left and right children (model 2)



# Generating the space of potential word order choices

- Both Lerner and Petrov (2013) and Jehl et al. (2014) make only *single-best* predictions
- We want:
  - *ALL REASONABLE* predictions instead of *SINGLE BEST*
  - More flexible model

# Producing multiple predictions

**Multiple predictions:**

- Bad: Mistakes in order decisions propagate

$\rightarrow$ Extract *n*-best decisions from the model to pass to subsequent model

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

- Preordered $\mathrm{s}$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

- Preordered $\mathrm{s}$
- Source dep. tree

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

- Preordered $\mathrm{s}$
- Source dep. tree
- Heads of all families

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

- Preordered $\mathrm{s}$
- Source dep. tree
- Heads of all families
- Local permutation

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathbf{s}' \mid \mathbf{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathbf{s}, h, \tau)$$

$$P_T(\pi \mid \mathbf{s}, h, \tau) \;=\; P(\psi \mid \mathbf{s}, h, \tau) \quad P_L(\pi_L \mid \mathbf{s}, h, \tau) \quad P_R(\pi_R \mid \mathbf{s}, h, \tau)$$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

$$P_T(\pi \mid \mathrm{s}, h, \tau) = P(\psi \mid \mathrm{s}, h, \tau) \quad P_L(\pi_L \mid \mathrm{s}, h, \tau) \quad P_R(\pi_R \mid \mathrm{s}, h, \tau)$$

- Pivot decision

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

$$P_T(\pi \mid \mathrm{s}, h, \tau) = P(\psi \mid \mathrm{s}, h, \tau) \; P_L(\pi_L \mid \mathrm{s}, h, \tau) \; P_R(\pi_R \mid \mathrm{s}, h, \tau)$$

- Pivot decision → $P(\psi \mid \mathrm{s}, h, \tau)$
- Left order decision → $P_L(\pi_L \mid \mathrm{s}, h, \tau)$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

$$P_T(\pi \mid \mathrm{s}, h, \tau) = P(\psi \mid \mathrm{s}, h, \tau) \; P_L(\pi_L \mid \mathrm{s}, h, \tau) \; P_R(\pi_R \mid \mathrm{s}, h, \tau)$$

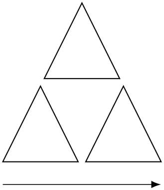
- Pivot decision → $P(\psi \mid \mathrm{s}, h, \tau)$
- Left order decision → $P_L(\pi_L \mid \mathrm{s}, h, \tau)$
- Right order decision → $P_R(\pi_R \mid \mathrm{s}, h, \tau)$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

$$P_T(\pi \mid \mathrm{s}, h, \tau) \;=\; P(\psi \mid \mathrm{s}, h, \tau) \quad P_L(\pi_L \mid \mathrm{s}, h, \tau) \quad P_R(\pi_R \mid \mathrm{s}, h, \tau)$$



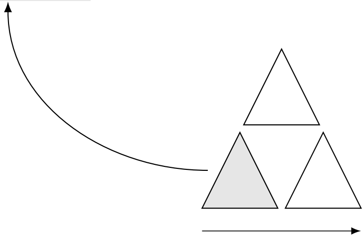
$$P(\mathrm{s}' \mid \mathrm{s}, \tau)$$

# Producing multiple predictions

**Model over possible orders of source words:**

$$P(\mathrm{s}' \mid \mathrm{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathrm{s}, h, \tau)$$

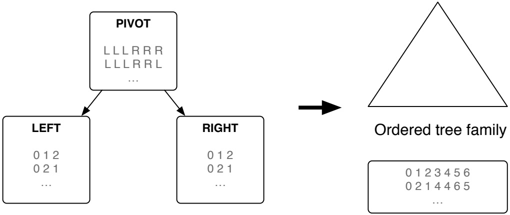
$$P_T(\pi \mid \mathrm{s}, h, \tau) = P(\psi \mid \mathrm{s}, h, \tau) \quad P_L(\pi_L \mid \mathrm{s}, h, \tau) \quad P_R(\pi_R \mid \mathrm{s}, h, \tau)$$



$$P(\mathrm{s}' \mid \mathrm{s}, \tau)$$

# Preordering alogrithm

- Produce $k_P$ best pivot decisions for all the children in the family
- For every of the $k_P$ pivot decisions:
  - Produce $k_L$ best left order decisions
  - Produce $k_R$ best right order decisions



# Preordering with arbitrary non-local features

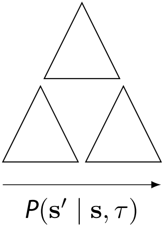
**Making the model more flexible:**

- Bad: Order decisions are local to tree families
- Khalilov and Sima'an (2012) show even weak LM helps with shortcomings

# Preordering with arbitrary non-local features

**Decoding:**

- Non-local features ruin our day...
- Cube pruning to the rescue (Chiang, 2007)!



# Preordering with arbitrary non-local features

**Preordering model:**

- Standard log-linear model (Och and Ney, 2002):

$$\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} \sum_i \lambda_i \log \phi_i(\mathbf{s}')$$

- Where to get the weights?
  - PRO: *tuning as ranking* (Hopkins and May, 2011)
  - Scoring functions:
    1. Kendall's $\tau$ coefficient
    2. Simulate word level MT system, score by BLEU

# Preordering with arbitrary non-local features

**Local features:**

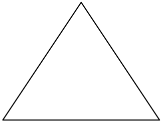
- Lexicalized preordering model $P(\mathrm{s}' \mid \mathrm{s}, \tau)$ from before
- Unlexicalized preordering model $P_W(\pi \mid h, cs)$ as less sparse backoff

**Non-local features:**

- ngram language models over $\mathrm{s}'$
  - words
  - part-of-speech tags
  - word classes

# Applicability of this model

- General model is applicable to any *n*-best preordering model over source trees
- **Example:**
  - Preordering model:
    Pairwise neural network-based model
    (de Gispert et al., 2015)
  - Parsing algorithm:
    *k*-best ITG-based CKY parsing
    (similar to Tromble and Eisner (2009)).



Ordered tree family

0 1 2 3 4 5 6
0 2 1 4 4 6 5
...

# Intrinsic: Do non-local features help?

- Intrinsic evaluation of preordering quality
- Language pair English-to-German

| Model | Kendall's tau | BLEU ($\hat{s}' \rightarrow s'$) |
|---|---|---|
| First-best $-$LM | 92.16 | 68.1 |
| First-best $+$LM (cube) | 92.27 | 68.7 |

# Translation: Quality of potential word order choices

- Translation experiments with the space of word order choices
- Experiments with top 10 preordering outputs of this model

| | Distortion | BLEU | MTR | TER |
|---|---|---|---|---|
| Baseline | 7 | 15.20 | 35.43 | 66.62 |
| Best out of *k* ($k = 10$) | | 17.26* | 37.97* | 62.64 |

# Discussion

**Preordering with non-local features**

- Integration of LM helps improve preordering quality
  - Slight Kendall $\tau$ improvement
  - BLEU preorder score shows benefits mostly in small local windows

**Quality of the space of potential word order choices**

- Experiments show significant potential improvement contained in the space
- With arbitrary *n* or lattice, space is small enough to be handled by subsequent models

# Conclusion

- Source preordering has big limitations but has proven very successful
- Our interest: Source-side adaptation models more suitable for morphologically rich languages

- First steps towards this goal:
  - Introduced preordering model that can delimit space instead of first-best predictions
  - More flexible model with arbitrary non-local features and cube pruning

# Thank You!

Any questions?

# References


Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

de Gispert, A., Iglesias, G., and Byrne, W. (2015). Fast and accurate preordering for SMT using neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2015)*.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jehl, L., de Gispert, A., Hopkins, M., and Byrne, B. (2014). Source-side preordering for translation using logistic regression and depth-first branch-and-bound search. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 239–248, Gothenburg, Sweden. Association for Computational Linguistics.

Khalilov, M. and Sima'an, K. (2012). Statistical translation after source reordering: Oracles, context-aware models, and empirical analysis. *Natural Language Engineering*, 18:491–519.

Lerner, U. and Petrov, S. (2013). Source-side classifier preordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Seattle, Washington, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA. Association for Computational Linguistics.

# References (cont.)

Tromble, R. and Eisner, J. (2009). Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore. Association for Computational Linguistics.