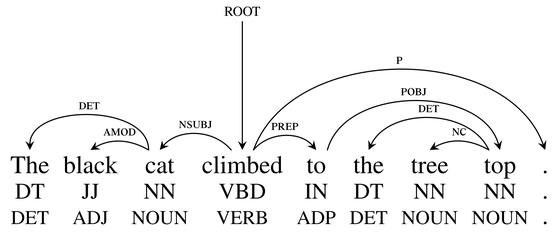
Source-Side Classifier Preordering for Machine Translation

Uri Lerner
Google, Mountain View

Slav Petrov
Google, New York

Preordering

Change source-side word order to make it more similar to target-side word order. Can improve word alignments and can be combined with any translation system.

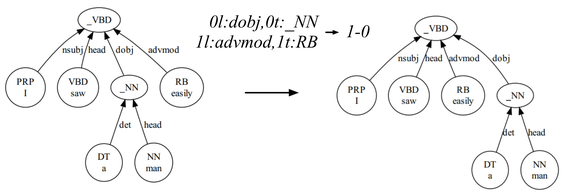


→ In Spanish order: *The cat black climbed to the top tree .*

→ In Japanese order: *The black cat the tree top to climbed .*

Related Work: Tree Transformations

Apply a set of transformations rules to parse trees.
Rules can be manually written or learned from data.



Reordering example from Genzel (2010), where the system learned to move RB over NN_.

When multiple rules apply: Use All? Most Frequent? Most Specific?
(worrisome 'Most Frequent' and 'Most Specific' are at opposite ends)

Related Work: Precedence Ranking

Assign a precedence score to each clause, then sort.

Tag	(Label, Weight, Order)
VB*	(advcl, 1, NORMAL) (nsubj, 0, NORMAL) (prep, 0, NORMAL) (dobj, -1, NORMAL) (prt, -2, REVERSE) (aux, 2, REVERSE) ...

Precedence rules from Xu et. al (2009). Rules were extracted manually by a bilingual speaker.

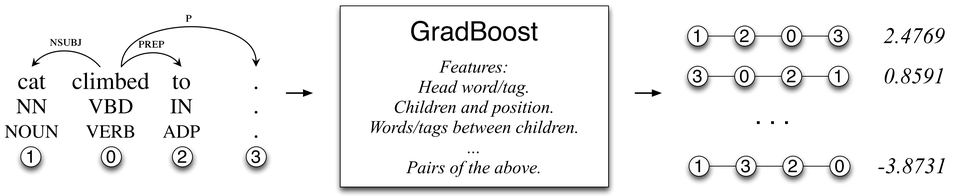
Precedence score computed independently for each clause/word:
No global consistency and hard to express priors
(e.g., A-B-C-D and D-C-B-A are likely but C-A-D-B is not).

Classifier Preordering

Predict the target word order by treating each permutation as a label in a multi-class classifier. Traverse the parse tree, reordering each family (head and children) and recursing.

Each classifier is trained on up to 15M training instances extracted from automatically aligned data. Limit to 20 most common permutations for ≥ 4 words.

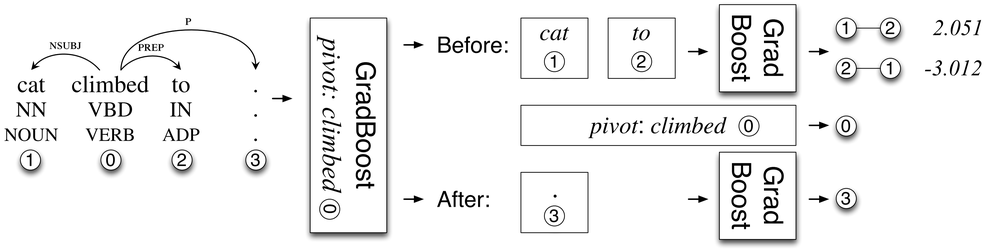
1-Step Classifier



Separate classifiers for 2, 3, …, 7 involved words.

2-Step Classifier

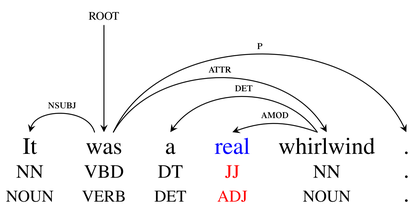
Decompose the search space: first determine the position of every child relative to the head (pivot) and then order the children before and after the head. Think QuickSort without recursion.



One pivot classifier and separate classifiers for 2, 3, …, 6 words before and after the pivot.
Only 1.07% of cases involve ≥ 5 words in the 2-step approach (vs. 5.54% for 1-step).
Pivot classifier has an accuracy of ≥ 95%.

Analysis

Some reordering decisions depend heavily on lexical choice.

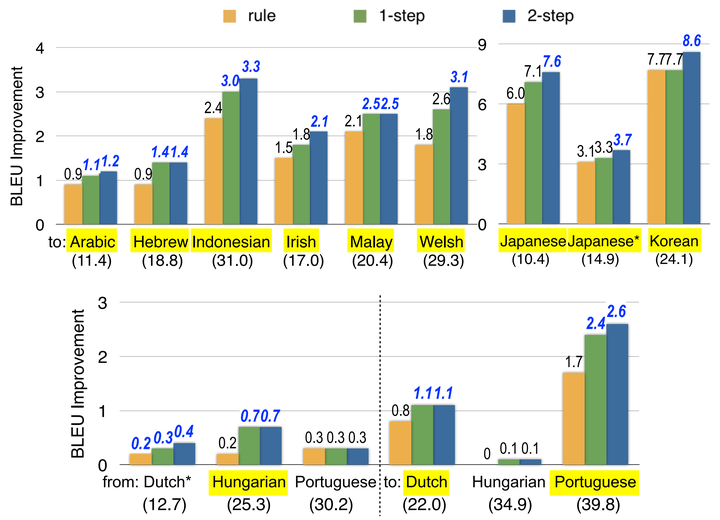


Feature	Weight
PrevChild:tag=JJ,PrevSibling:a	0.448
PrevChild:cat=ADJ,PrevSibling:a	0.292
PrevChild:cat=ADJ,NoNextSibling	0.212
...	
PrevChild:real,NoNextHeadSibling	-0.310
PrevChild:real,PrevSibling:cat=DET	-0.516
PrevChild:real,PrevSibling:a	-0.979

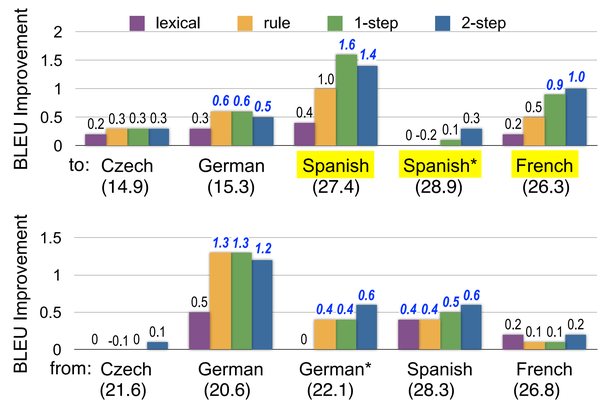
While adjectives typically follow nouns in Spanish, our classifiers have learned that '***real***' behaves differently and typically precedes nouns.

Results

Improvements relative to a phrase-based baseline for 1-step and 2-step classifier preordering and the system of Genzel 2010 (rule).



WMT 2010



* denotes a forest-to-string system and is shown only when better than the phrase-based system.
Improvements in ***blue*** are statistically significant at 99%.
For languages in yellow classifier reordering is better than rule.

Conclusions

- New approach: Directly predict in the exponential permutation space.
  - Decompose space via 2-step approach.
  - Limit to 20 most likely outcomes.
- Provides an elegant framework to combining different "rules."
  - Treat "rules" as features in a discriminative classifier.
- Can naturally express global consistency as priors.

Research at Google