Training a Natural Language Generator from Unaligned Data

Ondřej Dušek and **Filip Jurčíček** {odusek,jurcicek}@ufal.mff.cuni.cz
Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

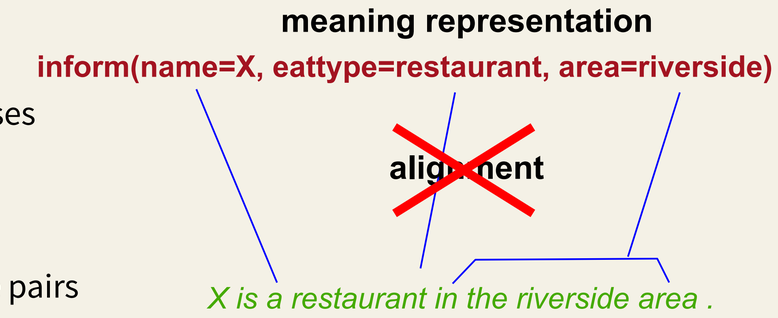
Learning NLG Without Alignments

Current NLG systems:

a) require alignment of MR elements to words/phrases
b) use a separate alignment step

Our generator:

• Learns alignment **jointly**, using just MR + Sentence pairs



Why?

• no need for manual annotation/preprocessing → cheaper/faster, no error propagation
• alignments are latent: MR elements do not need to correspond to words/phrases 1:1

Scorer

a function:

sentence plan tree + MR → real-valued score

• describes the fittness of the tree for the MR

Perceptron scorer

score = weights · features (from tree and MR)

• **features =** elements of tree and MR
  • presence of nodes, slots, values + combination
  • tree size, shape, parent-child
• **training** loop:
  • given MR, generate the best tree with current weights
  • update weights if generated tree ranks better than gold tree

Sentence Planner Algorithm

A*-style search:

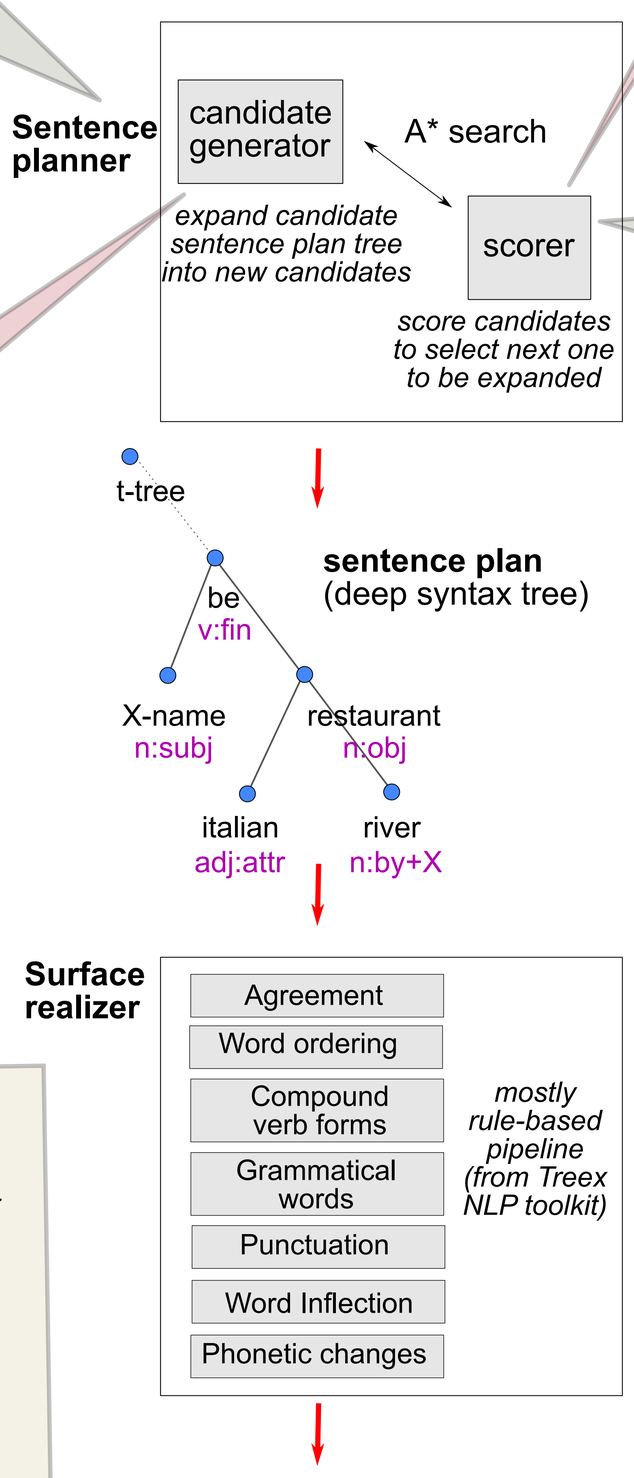
• "finding the path" from empty tree to full sentence plan tree
• expand the most promising candidate sentence plan in each step
• stop when candidates don't improve for a while

Two sub-components:

• **candidate generator** – do the expansion
• **scorer**/ranker – decide what's promising

Overall Schema of Our Generator

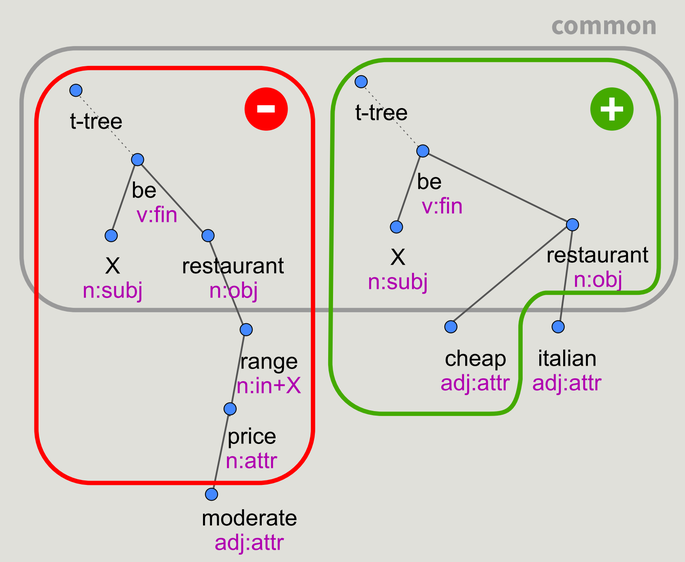
meaning representation (dialogue acts)
inform(name=X, type=placetoeat, eattype=restaurant, area=riverside, food=Italian)



Scorer Improvements

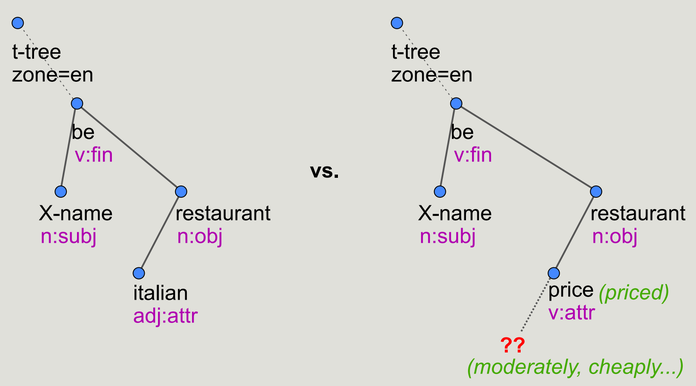
• features are global → bigger trees tend to score higher
• but we score incomplete trees
  • bigger is not always right
  • we must **promote the "promising"** ones
• two improvements to address this:

1) Differing subtree updates



• additional perceptron updates using pairs of differing subtrees from gold and generated tree

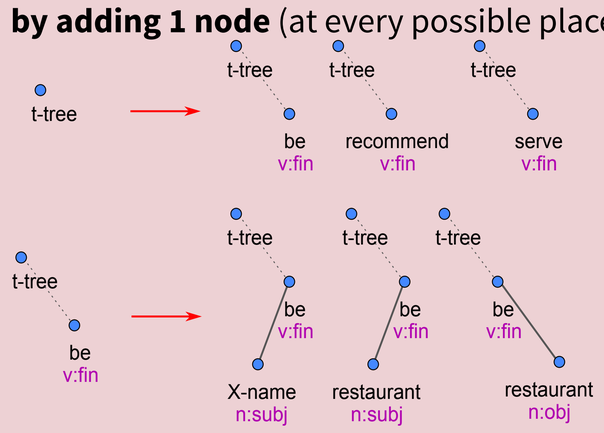
2) Future promise estimate



• estimate expected number of children for node type
• count how many nodes in a tree are missing to meet expectation of all nodes

Candidate Generator

• create **expansions** of the sentence plan tree **by adding 1 node** (at every possible place)



• "possible places" limited by a few simple context rules

Experiments

• data: **restaurant recommendation** from the *BAGEL* generator (Mairese et al., 2010)
  • restaurant location, food type, etc.
  • 404 sentences for 202 input dialogue acts, 2 paraphrases each
  • manual alignment provided, but we don't use it
• using 10-fold cross-validation
• measuring **BLEU/NIST**

Results

Setup	BLEU	NIST
perceptron scorer	54.24	4.643
+ differing subtree updates	58.70*	4.876
+ future promise	59.89*	5.231

• scorer improvements statistically significant
• lower scores than Mairesse et al.'s ~ 67% BLEU
• but our problem is harder:
  • we learn alignments jointly
  • our generator has to decide when to stop (whether all required information is included)

Example outputs

Input DA	inform(name=X-name, type=placetoeat, pricerange=moderate, eattype=restaurant)
Reference	X is a restaurant that offers moderate price range.
Generated	X is a restaurant in the moderate price range.
Input DA	inform(name=X-name, type=placetoeat, area=X-area, pricerange=moderate, eattype=restaurant)
Reference	X is a moderately priced restaurant in X.
Generated	X is a restaurant in the X area.
Input DA	inform(name=X-name, type=placetoeat, eattype=restaurant, area=riverside, food=French)
Reference	X is a French restaurant on the riverside.
Generated	X is a French restaurant in the riverside area which serves French food.

Conclusion

• Learning sentence planning from unaligned data is feasible
• Promising results, but lower than previous with manual alignment (Mairesse et al., 2010)
  • outputs mostly fluent and relevant (with some problems)

Our generator is available at:

http://github.com/UFAL-DSG/tgen

Based on Dušek & Jurčíček, ACL 2015. Presented at YRRSDS, September 2015, Prague.
This work was funded by MEYS Czech Rep. grant LK11221 and core research funding, SVV project 260 104, and GAUK 2058214 of Charles Univ. in Prague. It uses language resources from the LINDAT/CLARIN project , MEYS Czech Rep. grant LM2010013.