Reading about Search

Ondřej Bojar

May 2, 2019

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Outline

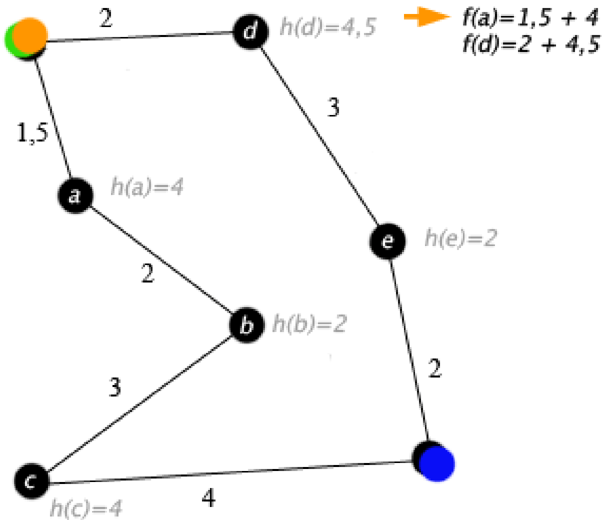
- Intro: Dijskra and A* search.
- MT is NP-hard.
- Fast and optimal decoding.
- Stacks and future cost.
- Cube pruning.
- Hypergraph decoding.

Dijkstra and A* Search

- Dijkstra's algorithm for shortest path:
  - Always extend the cheapest/shortest option.
- A* (A-Star) Search:
  - Always extend the cheapest/shortest option.
  - Include a CONSISTENT (optimistic) heuristic estimate of the remaining distance (also called "future cost").

Key data structure: stack of open hypotheses.

A* Search



Dijkstra		A*	
a	1.5	a	$+4 = 5.5$
d	2.0	b	$+2 = 5.5$
b	3.5	d	$+4.5 = 6.5$
e	5.0	e	$+2 = 7.0$
c	5.5	goal	$+0 = 7.0$
goal	7.0	c	$+4 = 9.5$

MT is NP-Hard

NP-hard problem:

- To solve the task of size n, strictly more than n^k steps (for any fixed k) have to be made.
- Usually this means, there are exponetially (k^n) solutions to consider.

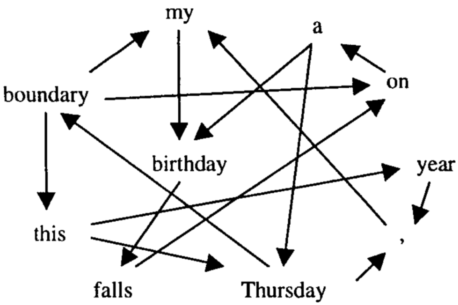
Knight (1999) shows word-based MT is NP-hard for two reasons:

- Selecting source word order ($\rightarrow$ Hamilton circuit).
- Grouping source words to form multi-word dictionary entries ($\rightarrow$ Minimum set cover).

- These are worst-case constructions.

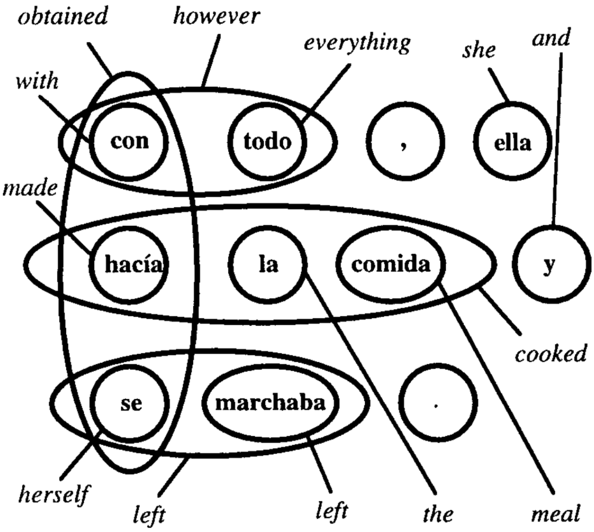
MT is NP-Hard (2/3)

- Remember the NP-hardness proof strategy:
  - Use MT as a black box to solve an NP-complete task.



With a 2-gram language model, finding the best word ordering solves the Hamilton Circuit or Travelling Salesman Problem. (Knight, 1999)

MT is NP-Hard (3/3)



Selecting a set of multi-word translations to cover the whole sentence solves Minimum Set Cover Problem. (Knight, 1999)

Input: However, she cooked and left.

Fast and Optimal Decoding

Germann et al. (2004) implement three word-based decoders:

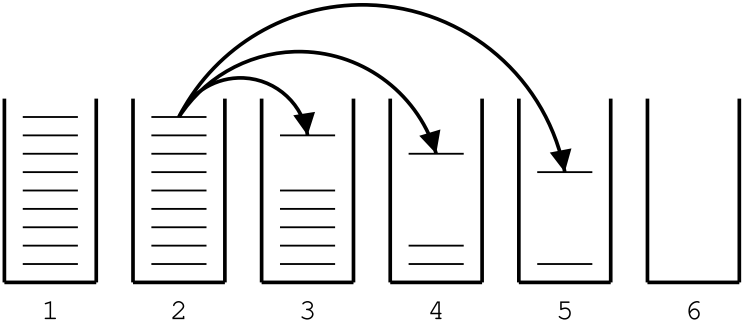
- Stack-based.
  - Similar to Moses but 2^n stacks instead of n stacks.
- Greedy.
  - Start with the cheapest gloss.
  - Modify alignment and translation to improve probability.
- Optimal ($\sim$Traveling Salesman).
  - Finding a tour through all source cities gives us target translation by noting owners of hotels where we stayed.

Observations:

- Many pure modelling errors.
- Greedy decoding viable option.

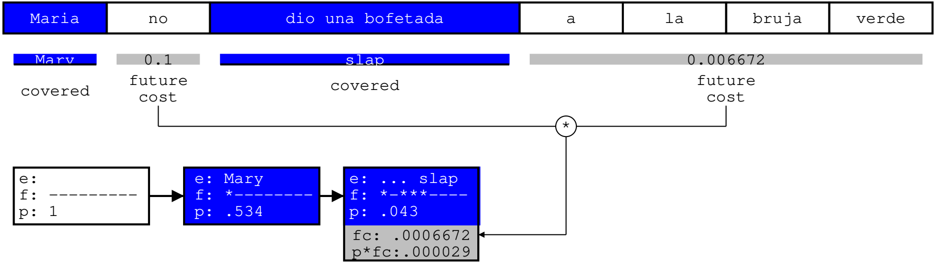
Stacks and Future Cost (1)

Remember Moses/Pharaoh stack-based decoding:



- n stacks based on number of words covered.
- A stack contains hypotheses regardless which words were covered.
  $\Rightarrow$ Not a fair comparison.

Stacks and Future Cost (2)



- Future cost to make the competition fair.
- Future cost = consistent heuristic estimate.
  Optimistic, because LM will make attachments more expensive.
- No future cost would be needed, if stacks were infinite.

Reranking

"Feature engineering":

- Choosing the most promising hypotheses based on local observations.

Some features need more context of output hypotheses, e.g.:

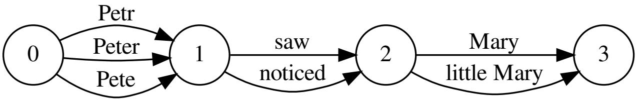
- Is the output hypothesis syntactically correct?
  $\Rightarrow$ Need full parsing.

Reranking example:

1. Generate n-best list of hypotheses.
2. Parse all of them.
3. Prefer hypotheses with likely parses.

Local vs. Non-local Features

Non-local features facilitate reranking of partial hyps. (Lopez, 2009)
While building partial hypotheses, decisions multiply:



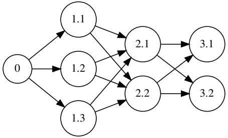
$3 \cdot 2 \cdot 2 = 12$ hyps.
$\Rightarrow$ n-best lists inevitably too short.

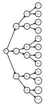
Local features access only **input** and **current edge**:

- Do we prefer to translate "Petr" as "Peter" or leave non-translated?

Non-local features access **partial history**:

- Do we prefer "Pete saw" or "Peter noticed"?
- Can be seen as **state splitting** depending on the relevant past context.





Reranking can access **full history**.

Weighted Deduction

Lopez (2009) summarizes several decoders in a unified framework of weighted deduction:

- Left-to-right, phrase-based, CKY.
- A HYPERGRAPH (see e.g. Huang (2008)) represents the deductions: combining items according to deduction rules.

Non-local features:

- Accommodated by state splitting ("product" of logics).

See the slides by Adam Lopez.

Cube Pruning

- Only a fraction of hypotheses constructed will escape pruning.
- Let's not construct them at all!
- Instead: Construct elements of the product starting from the (approximately) cheapest until the target stack is full.
- More details in Huang and Chiang (2007).

Summary

- MT is NP hard.
- Sub-optimal algorithms (stack-based, greedy, ...) used.
    - Modelling errors are an issue.
    - Future cost to reduce search errors.
- Local and non-local features.
- Unified view: translation as weighted deduction:
    - State splitting.
    - Cube pruning for stack-based decoding.

References

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2004. Fast and optimal decoding for machine translation. Artif. Intell., 154(1-2):127–143.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.

Liang Huang. 2008. Forest-Based Algorithms in Natural Language Processing. Ph.D. thesis, University of Pennsylvania.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. Comput. Linguist., 25(4):607–615.

Adam Lopez. 2009. Translation as weighted deduction. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 532–540, Athens, Greece, March. Association for Computational Linguistics.