Human Language Technology:
Applications to Information Access

Lesson 8: Decoding for MT: Beam search stacked decoding

November 10, 2016

EPFL Doctoral Course EE-724
Andrei Popescu-Belis
Idiap Research Institute, Martigny

Reminder: statistical MT

- Principle (using Bayes' theorem)
  - learn English **language model**: $P(e)$
  - learn (reverse) **translation model**: $P(f|e)$
  - **decode** source sentence: find most likely *e* given *f*

$$\text{argmax}_e\ P(e|f) = \text{argmax}_e\ (P(f|e)\ P(e))$$

- Given a generative model, how to find the best translation?
  - brute force search is a theoretical (but impractical) solution
  - decoding algorithms: **find argmax $P(e|f)$ quickly and reliably**

Phrase-based translation probability

$$P_{\text{TM}}(f|e) = \Pi_{i=1..M}\, P(\underline{f}_i|\underline{e}_i) \cdot d(\text{START}(\underline{f}_i) - \text{END}(\underline{f}_{i-1}) - 1)$$

$P(\underline{f}_i|\underline{e}_i)$ is the prob. that phrase $\underline{e}_i$ is translated into $\underline{f}_i$

d is a "distance-based reordering model" (e.g. $\alpha^{|x|}$)

START($\underline{f}_i$) is the position of the first word of phrase $\underline{f}_i$

END($\underline{f}_{i\text{-}1}$) is the position of the last word of phrase $\underline{f}_{i\text{-}1}$

- e.g,, if phrases $\underline{e}_{i\text{-}1}$ and $\underline{e}_i$ are translated in sequence by phrases $\underline{f}_{i-1}$ and $\underline{f}_i$, then START($\underline{f}_i$) = END($\underline{f}_{i-1}$)+1, and we have $d(0)$

Log-linear models

- Translate *f* = find *e* which maximizes the product of 3 types of terms
  - probabilities of inverse phrase translations $P_{tm}(\underline{f}_i|\underline{e}_i)$
  - reordering model for each phrase $d(\text{START}(\underline{f}_i) - \text{END}(\underline{f}_{i-1}) - 1)$
  - language model for each word $P_{lm}(e_k|e_1,\ldots, e_{k-1})$
- Terms can be weighted: no longer Bayesian model, but more efficient
  - more components can be added + weights can be tuned
- So, now we want to find the sentence that maximizes

$$\prod_{i=1..M} (P_{tm}(\underline{f}_i|\underline{e}_i)^{\lambda_{tm}} \cdot d(\text{START}(\underline{f}_i)-\text{END}(\underline{f}_{i-1})-1)^{\lambda_{rm}})\cdot\prod_{k=1..|e|} P_{lm}(e_k|e_1,\ldots, e_{k-1})^{\lambda_{lm}}$$

which can be expressed as: $\exp(\sum \lambda_i h_i(e))$ using $h(\ldots) = \log P(\ldots)$

and is thus equivalent to maximizing the sum without the 'exp'

Intuitive view of searching

- Given a foreign sentence f
- Pick a word or a phrase f_1 to translate
  - e.g. from the beginning, but not necessarily
  - get from the phrase table a possible translation $\underline{e}_1$
  - put it *at the beginning* of the e sentence
- Pick a second phrase f_2 to translate
  - not necessarily after f_1
  - get a possible translation $\underline{e}_2$
  - put it *just after* $\underline{e}_1$ in the e sentence
- …
- Continue this process until there are no phrases left to translate from the source sentence f
  - obtain a complete translation e

➔ All these operations have a cost, which is used to score e

Scoring translation hypotheses *e*

- Cost of building a translation hypothesis
  - the smaller the probability, the larger the cost
  - in the log-linear model, costs are additive and weighted

- Components of the cost
  - translation model (λ_{tm}): probability from phrase table
  - reordering model (λ_d): distortion probability can be modeled as $d(\text{START}(\underline{f}_i) - \text{END}(\underline{f}_{i-1}) - 1)$, based on the position of the previous phrase $\underline{f}_{i-1}$
  - language model (λ_{lm}): in a n-gram model, based on the previous n–1 words

- We can estimate the cost of a partial or a complete hypothesis → the goal is to find the lowest cost one

Towards realistic search strategies

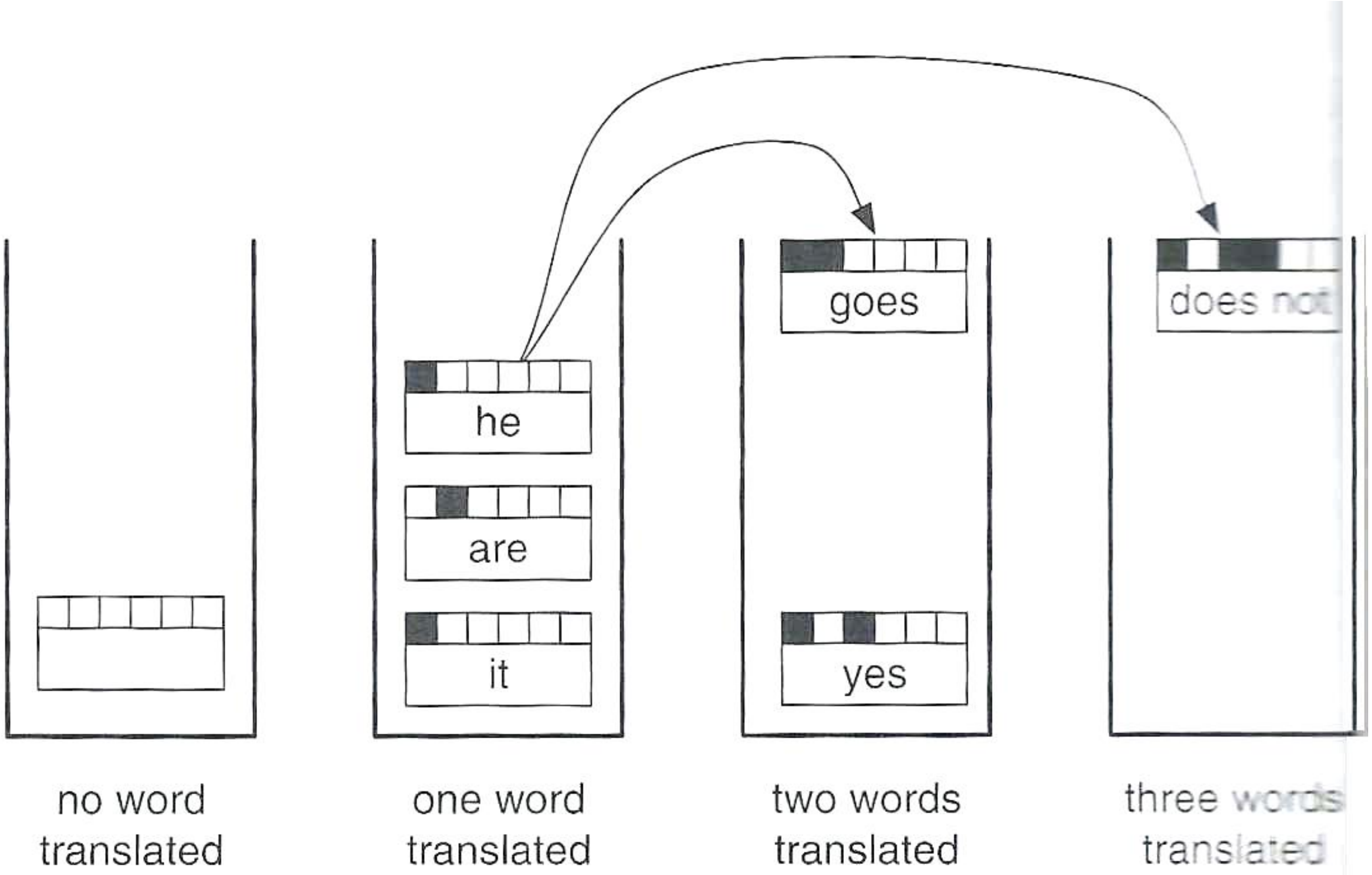
- Starting from the beginning of the target sentence (as on slide 5), increment it in order, by considering all possible translations from the phrase table, until no source phrase is left untranslated
  - these hypotheses form a search graph
  - an end point in the graph = a complete translation hypothesis
  - the goal is to find the lowest cost path in the graph

- Search space grows exponentially with sentence length
  - i.e. decoding is NP-complete
  - heuristics to reduce search space
    - hypothesis recombination
    - pruning by organizing hypotheses into stacks
      - prune stacks based on stack size and cost
      - set a maximum reordering limit
      - prune based on future cost estimation

Hypothesis recombination

- When two (partial) hypotheses lead to the same “state”, the more expensive one can be deleted
  - because it certainly won’t lead to a cheaper translation
  - “state” means
    - last *k* words, where *k* is the order of the LM
    - position of last phrase and last-but-one (reordering model)
    - same last phrase
- This simplifies search with no risk of missing the best translation – but complexity remains exponential

Hypothesis stacks: group hypotheses by number of translated words (Koehn 2010, page 164)



Pruning stacks

- Histogram pruning
  - keep at most *n* hypotheses in each stack

- Threshold pruning
  - keep hypotheses with a cost no worse than *X*% of the best currently found one
    - 1–*X* is the size of the **beam**

- The use of pruning
  - in practice: combine both methods of pruning
  - no longer guarantees finding the best translation
  - reduces complexity from exponential to quadratic
    - *O*(max_stack ·× nb_of_options × sentence_length)

Limiting the reordering

- Additional constraint on hypothesis expansion

- When choosing source phrase $\underline{f}_i$ to generate target phrase $\underline{e}_i$, limit the difference between START($\underline{f}_i$) and END($\underline{f}_{i-1}$) to d_{max} words (e.g. $d_{\mathrm{max}} \leq 5$)

- Decreased complexity
  - the number of possible expansions of each hypothesis no longer grows with sentence size
  - O(max_stack × sentence_length) … which is linear

Estimating the future cost of a hypothesis

- Goal: for a given hypothesis, quickly estimate the difficulty of what remains to be translated
  - add this cost to the cost of each partial hypothesis when pruning each stack
  - doing so avoids keeping only hypotheses that translate the easiest part of a sentence (among all those of length *n*)

- How to estimate future cost? (No magic allowed.)
  - for one translation option to grow a hypothesis:

    translation model (= lookup phrase in table)
    + language model (= apply only over the new phrase)
    + reordering (… ignored)

  - for an entire remaining span: find the cheapest coverage with phrases, using a dynamic programming method

Summary

- Perform the following operations until no new translation hypotheses can be created
- For each stack
  - for each hypothesis
    - grow hypothesis in several ways
    - place resulting hypotheses in respective stack
      - if possible, recombine with an existing hypothesis
      - estimate future costs
      - prune the stack if too big (max size and beam size)
- The complete hypothesis (all words translated) with the highest score (lowest cost) is the result

Conclusion

- One main story but with a lot of variants and additions
  - for translation modeling and for decoding
  - major alternative: syntax-based approach (tree-based)
  - factored models allow using additional constraints
- The three building blocks of MT are now into place in the course
  - LM learning | TM learning | decoding
  - one missing block: evaluation methods (next time)
- Practical work: see TP-MT-instructions
  - training and decoding with Moses
- References: Philipp Koehn, *Statistical Machine Translation*, Cambridge University Press, 2010, Chapter6; Kevin Knight, “Decoding complexity in word-replacement translation models”, Computational Linguistics, vol. 25, n. 4, p. 607-615, 1999.