Administrative

- Homework 2: due a week from tomorrow
- Leaderboard will be up soon
Big Picture

- pre-processing
- alignment
- model extraction

- tuning
- decoding
- post-processing

Today

- scoring

training → translation → evaluation
However, the sky remained clear under the strong north wind.
Decoding Review

• Given a model, we want to find the solution to

\[ e^* = \underset{e}{\text{argmax}} \ p(e \mid f) \]

• Dynamic programming provides the approximation

\[ (e^*, a^*) = \underset{e, a}{\text{argmax}} \ p(e, a \mid f) \]
Factored search

- Translate a word (or phrase) at a time
- Assemble English translation left-to-right
- Maintain a data structure that records these

- Sentence: *Yo tengo hambre*

  \[
  \text{generated English} \\
  \begin{array}{c}
  <s> \text{I} \\
  \end{array}
  \begin{array}{c}
  \text{tengo} \\
  \end{array}
  \rightarrow
  \begin{array}{c}
  \text{am} \\
  \end{array}
  =
  \begin{array}{c}
  <s> \text{I am} \\
  \end{array}
  \]

  coverage vector
- Example hypothesis creation:

\[
\text{old hypothesis} \quad + \quad \text{add word} \quad = \quad \text{new hypothesis}
\]

- translation model: trivial case, since all the words are translated independently

\[
\text{hypothesis.score} += \log P_{TM}(\text{am} \mid \text{tengo})
\]

- a function of just the word (or phrase) that is added
- Example hypothesis creation:

\[
\begin{align*}
<s> \text{I} & \quad + \quad \text{tengo} \rightarrow \text{am} \\
\text{old hypothesis} & \quad \text{add word} \\
\text{new hypothesis}
\end{align*}
\]

- language model: still easy, since (bigram) language models depend only on the previous word

\[
\text{hypothesis.score} += P_{LM}(\text{am} | \text{I})
\]

- a function of the old hyp. and the new word translation
• These items can then be arranged into a **search graph** representing the whole translation space

• Component models need to factorize over this graph
Yo tengo hambre
Algorithm

- Start with a list containing the empty hypothesis

- Loop
  - For each hypothesis
    - For each remaining untranslated word
      - Extend the hypothesis with the word
      - Add to the list of hypotheses
Demo

Full search, no dynamic programming
Dynamic programming

- All submodels factorize into lattices where only local information is needed

- **Recombination**: chart items that are the same with respect to dynamic programming state can be combined

- Backpointers allow the full translation to be recovered
- Notice anything here?

- (1) `<s>` is never used in computing the scores AND (2) `<s>` is implicit in the graph structure

- Why are we lugging these old words around?

\[
\text{score += } P_{TM}(am | \text{ tengo}) + P_{LM}(am | I)
\]
The score of the new hypothesis is the highest-scoring way to produce it.
Dynamic Programming

• Start with a list containing the empty hypothesis

• Loop

  • For each hypothesis

    • For each remaining untranslated word

      • Extend the hypothesis with the word

      • Add to the list of hypotheses if (a) no equivalent item exists or (b) this one has a higher score
Demo

Dynamic programming
DP State

• What is an “equivalent hypothesis” (or item)?
  
  • An item that matches on all shared state
    
    • coverage vector
    
    • previous word (needed for LM)
  
  • In general, the items will also cache the score of the Viterbi path to that point, and maintain a list of pointers to all incoming tail nodes
Stack decoding

• There are still too many hypotheses

• Also, all hypotheses are in competition:
  • Search expansion that always extends the lowest-scoring item will waste lots of time

• Stack decoding groups items by coverage vectors
Stack decoding

• Start with a list containing the empty hypothesis

• Loop

  • For each stack

    • For each hypothesis

      • For each remaining untranslated word

        • Extend the hypothesis with the word

        • Add hypothesis to the right stack group
Pruning

• There are still too many hypotheses

• We can prune the stacks in many ways:
  
  • **Histogram pruning**: limit stacks to size $N$
  
  • **Threshold pruning**: remove hypotheses with a score $\varepsilon$ below the highest-scoring item in the stack
Stack decoding

With pruning

• Start with a list containing the empty hypothesis

• Loop

  • For each stack

    • For each hypothesis

      • For each remaining untranslated word

        • Extend the hypothesis with the word

        • Add hypothesis to the right stack group

      • Prune
Demo

“Stack” decoding with beam search
Distortion limits

• There are still too many hypotheses

• Distortion limits restrict the set of source language words that can be used to extend a hypothesis to those with $d$ words
Stack decoding

With pruning, distortion limits

- Start with a list containing the empty hypothesis

- Loop
  - For each stack
    - For each hypothesis
      - For each remaining *eligible* untranslated word
        - Extend the hypothesis with the word
        - Add hypothesis to the right stack group
      - Prune
• With these innovations, decoding has been reduced to linear time (in the sentence length)

• Types of errors:

  • search: approximations preclude model-best translation

  • model: highest-scoring translation is not the actual best translation
Important concepts

• Decoding as graph search
• Stack decoding and dynamic programming
• Factorized models for edge scoring
• Pruning (histogram, threshold)