Decoding

continued
Activity

Build a translation model that we’ll use later today.

Instructions

• Subject is “mt-class”
• The body has six lines
• There is one, one-word translation per line
- Schedule for language in 10 minutes

- Leaderboard
However, the sky remained clear under the strong north wind.
SCHEDULE

- TUESDAY
  - stack-based decoding in conception

- TODAY
  - stack-based decoding in practice
  - scoring, dynamic programming, pruning
Decoding

- the process of producing a translation of a sentence

- Two main problems:

  - modeling – given a pair of sentences, how do we assign a probability to them?

\[ P_{(C \rightarrow E)} \]

They still lack experience in international competitions

They still lack experience in international competitions

- high
- the process of producing a translation of a sentence

- Two main problems:

  - **modeling** – given a pair of sentences, how do we assign a probability to them?

  \[
P_{(C \rightarrow E)}(\text{他们还缺乏国际比赛的经验.}) = \text{low}
\]

  This is not a good translation of the above sentence.
- Noisy Channel model

\[ P(e | f) \propto P(f | e)P(e) \]
- Add weights

\[
P(e \mid f) \propto P(f \mid e)P(e) \\
\propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2}
\]
- Why?

- Just like in real life, where we trust people’s claims differently, we will want to learn how to trust different models.

“"I can do a backflip off this pommel horse"
- Log space transform

\[ P(e \mid f) \propto P(f \mid e)P(e) \]
\[ \propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2} \]
\[ = \lambda_1 \log P(f \mid e) + \lambda_2 \log P(e) \]

- Because:
\[ 0.0001 \times 0.0001 \times 0.0001 = 0.000000000001 \]
\[ \log(0.0001) + \log(0.0001) + \log(0.0001) = -12 \]
- Generalization

\[ P(e \mid f) \propto P(f \mid e)P(e) \]
\[ \propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2} \]
\[ = \lambda_1 \log P(f \mid e) + \lambda_2 \log P(e) \]
\[ = \lambda_1 \phi_1(f, e) + \lambda_2 \phi_2(f, e) \]
\[ = \sum_{i} \lambda_i \phi_i(f, e) \]
A better “fundamental equation” for MT

\[ e^*, a^* = \arg\max_{e,a} \Pr(e, a | c) \]
**Decoding**

- *the process of producing a translation of a sentence*

- Two main problems:
  
  - **search** – given a model and a source sentence, how do we find the sentence that the model likes best?

  - impractical: enumerate all sentences, score them

  - stack decoding: assemble translations piece by piece
- Start with a list of hypotheses, containing only the empty hypothesis

- For each stack

  - For each hypothesis

    - For each applicable word

      - Extend the hypothesis with the word

      - Place the new hypothesis on the right stack
FACTORYING MODELS

- Stack decoding works by extending hypotheses word by word

- These can be arranged into a *search graph* representing the space we search
FACTORY MODELS

Yo → I

I tengo → have

I am → am

hambre → hunger

I have hunger → have

I am hungry → am

hunger → am
FACTORYING MODELS

- Stack decoding works by extending hypotheses word by word

- These can be arranged into a search graph representing the space we search

- The component models we use need to factorize over this graph, and we accumulate the score as we go
**Factoring models**

- Example hypothesis creation:

```
<start> I
+ tengo → am
= <start> I am
```

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

```
hypothesis.score += P_{TM}(am | tengo)
```

- a function of just the word that is added
- **Example hypothesis creation:**

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

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

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

- a function of the old hyp. and the new word translation
- We saw Tuesday how huge the search space could get

- Notice anything here?

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

- let’s get rid of the extra state!

\[
\begin{align*}
\text{old hypothesis} & \quad + \quad \text{add word} \quad = \quad \text{new hypothesis} \\
\text{score} & \quad += \quad P_{TM}(am \mid tengo) \\
& \quad + \quad P_{LM}(am \mid I)
\end{align*}
\]
The score of the new hypothesis is the maximum way to compute it.
STACK DECODING (WITH DP)

- Start with a list of hypotheses, containing only the empty hypothesis

- For each stack

  - For each hypothesis

    - For each applicable word

      - Extend the hypothesis with the word

    - Place it on the right stack IF either (1) no equivalent hypothesis exists or (2) this hypothesis has a higher score.
MORE GENERALLY

- What is an “equivalent hypothesis”?

- Hypotheses that match on the minimum necessary state:

  - last word (for language model computation)

  - the score (of the best way to get here)

  - the coverage vector (so we know which words we haven’t translated)
OLD GRAPH (BEFORE DP)
- Even with DP, there are still too many hypotheses

- So we prune:

  - histogram pruning: keep only $k$ items on each stack

  - threshold pruning: don’t keep items that have a score beyond some distance from the most probable item in the stack
- Start with a list of hypotheses, containing only the empty hypothesis

- For each stack

  - For each hypothesis

    - For each applicable word

      - Extend the hypothesis with the word

      - If it’s the best, place the new hypothesis on the right stack (possible replacing an old one)

    - Prune
- **Search errors**
  - *def*: not finding the model’s highest-scoring translation
  - this happens when the shortcuts we took excluded good hypotheses

- **Model errors**
  - *def*: the model’s best hypothesis isn’t a good one
  - depends on some metric (e.g., human judgment)
Activity

http://cs.jhu.edu/~post/mt-class-stack-decoder/

Instructions (10 minutes)
In groups or alone, find the highest-scoring translation under our model under different stack size and reordering settings.

Are there any search or model errors?
- generalized weighted feature function formulation
- decoding as graph search
- factorized models for scoring edges
- dynamic programming
- pruning (histogram, beam/threshold)
NOT DISCUSSED (BUT IMPORTANT)

- Outside (future) cost estimates and A* search

- Computational complexity