Factored Language Models

EE517 Lecture
April 19, 2005
Kevin Duh (duh@ee.washington.edu)

Outline
1. Motivation
2. Factored Word Representation
3. Generalized Parallel Backoff
4. Model Selection Problem
5. Applications
6. Tools

Word-based Language Models
- Standard word-based language models
  \[p(w_1, w_2, ..., w_T) = \prod_{t=1}^{T} p(w_t | w_1, ..., w_{t-1})\]
- How to get robust n-gram estimates (\(p(w_1, w_2, ..., w_T)\))?
  - Smoothing
    - E.g. Kneser-Ney, Good-Turing
  - Class-based language models
    \[p(w_t | w_{t-1}) = p(w_t | C(w_t))p(C(w_t) | C(w_{t-1}))\]

Limitation of Word-based Language Models
- Words are inseparable whole units.
  - E.g. "book" and "books" are distinct vocabulary units
- Especially problematic in morphologically-rich languages:
  - E.g. Arabic, Finnish, Russian, Turkish
  - Many unseen word contexts
  - High out-of-vocabulary rate
  - High perplexity

Arabic Morphology
- Pattern
  - \(fa\- sakan\- tu\)
  - Affixes
- Root
  - \(LIVE\ +\ past\ +\ 1st-sg-past\ +\ part:\ "so\ I\ lived"\)
- \(~5000\ roots\)
- Several hundred patterns
- Dozens of affixes

Vocabulary Growth - full word forms
**Vocabulary Growth - stemmed words**

- Final Report from the JHU Summer Workshop 2002
- Source: K.

**Factored Word Representations**

\[ w = \{ f^1, f^2, \ldots, f^K \} \]

\[ p(w_1, w_2, \ldots, w_T) = p(f_1^{1:T}, f_2^{1:T}, \ldots, f_K^{1:T}) \]

\[ = \prod_{t=1}^{T} p(f_t^{1:K} | f_{t-1}^{1:K}) \]

- Factors may be any word feature. Here we use morphological features:
  - E.g. POS, stem, root, pattern, etc.

**Example**

- Training sentence: “Iazim tiqra kutubi bi sorca”
  (You have to read my book quickly)
- Test sentence: “Iazim tiqra kitAbiy bi sorca”
  (You have to read my book quickly)

**Language Model Backoff**

- When n-gram count is low, use (n-1)-gram estimate
- Ensures more robust parameter estimation in sparse data:

**Solution: Word as Factors**

- Decompose words into “factors” (e.g. stems)
- Build language model over factors: \( P(w|factors) \)
- Two approaches for decomposition
  - Linear
    - [e.g. Geutner, 1995]
  - Parallel
    - [Kirchhoff et. al., JHU Workshop 2002]
    - [Bilmes & Kirchhoff, NAACL/HLT 2003]

**Advantage of Factored Word Representations**

- Main advantage: Allows robust estimation of probabilities (i.e. \( P(f_1^{1:T}, f_2^{1:T}, \ldots, f_K^{1:T}) \)) using backoff
  - Word combinations in context may not be observed in training data, but factor combinations are
  - Simultaneous class assignment

**Word**

- Kitaab-iy
  - (My book)
- Kutub-iy
  - (Their book)

**Backoff**

- Drop most distant word during backoff

**Backoff graph: multiple backoff paths possible**

- Word-based LM:
- Factored Language Model:
Choosing Backoff Paths

- Four methods for choosing backoff path
  1. Fixed path (a priori)
  2. Choose path dynamically during training
  3. Choose multiple paths dynamically during training and combine result (Generalized Parallel Backoff)
  4. Constrained version of (2) or (3)

Generalized Backoff

- Katz Backoff:
  \[ P_a(w_t | w_{t-1}, s_{t-1}) = \frac{N(w_t, w_{t-1}, s_{t-1})}{N(w_t, w_{t-1})} \quad \text{if } N(w_t, w_{t-1}, s_{t-1}) > 0 \]
  \[ \alpha(w_t, w_{t-1}) P_a(w_t | w_{t-1}) \quad \text{otherwise} \]

- Generalized Backoff:
  \[ P_a(f_t | f_{t-1}, f_{t-2}) = \frac{N(f_t, f_{t-1}, f_{t-2})}{N(f_t, f_{t-1})} \quad \text{if } N(f_t, f_{t-1}, f_{t-2}) > 0 \]
  \[ \alpha(f_t, f_{t-1}) g(f_t, f_{t-1}) \quad \text{otherwise} \]

where \( g() \) can be any positive function, but some \( g() \) makes \( \alpha() \) difficult to compute.

g() functions

- A priori fixed path:
  \[ g(f, f_{t-1}, f_{t-2}) = P_{bo}(f | f_{t-1}) \]

- Dynamic path: Max counts:
  \[ g(f, f_{t-1}, f_{t-2}) = P_{bo}(f | f_{t-1}) \]
  \[ j^* = \arg \max \frac{N(f, f_{t-1})}{N(f)} \]
  
  Based on raw counts => Favors robust estimation

- Dynamic path: Max normalized counts:
  \[ j^* = \arg \max \frac{N(f, f_{t-1})}{N(f)} \]
  
  Based on maximum likelihood => Favors statistical predictability

Multiple Backoff Paths: Generalized Parallel Backoff

- Choose multiple paths during training and combine probability estimates

\[ P_b(w_t | w_{t-1}, s_{t-1}) = \frac{d_j P_a(w_t | w_{t-1}, s_{t-1})}{d_j} \quad \text{if } d_j \geq \text{threshold} \]

\[ = \frac{2}{d_j \text{no count}} \left( P_a(w_t | w_{t-1}, s_{t-1}) + P_b(w_t | w_{t-1}, s_{t-1}) \right) \quad \text{else} \]

Options for combination are:

- average, sum, product, geometric mean, weighted mean

Dynamically Choosing Backoff Paths During Training

- Choose backoff path based on \( g() \) and statistics of the data

Summary:
Factored Language Models

Factored Language Model =
Factored Word Representation + Generalized Backoff

- Factored Word Representation
  - Allows rich feature set representation of words
- Generalized (Parallel) Backoff
  - Enables robust estimation of models with many conditioning variables
Model Selection Problem

- In n-grams, choose, eg.
  - Bigram vs trigram vs 4gram
    => relatively easy search; just try each and note perplexity on development set
- In Factored LM, choose:
  - Initial Conditioning Factors
  - Backoff Graph
  - Smoothing Options
    => Too many options; need automatic search
    => Tradeoff: Factored LM is more general, but harder to select a good model that fits data well.

Example: a Factored LM

- Initial Conditioning Factors, Backoff Graph, and Smoothing parameters completely specify a Factored Language Model
- E.g. 3 factors total:
  0. Begin with full graph structure for 3 factors
  1. Initial Factors specify start-node
  2. Specify backoff graph: i.e. what backoff to use at each node
  3. Begin with subgraph obtained with new root node
  4. Specify backoff graph: i.e. what backoff to use at each node
  5. Specify smoothing for each edge

Applications for Factored LM

- Modeling of Arabic, Turkish, Finnish, German, and other morphologically-rich languages
  - [Kirchhoff, et. al., JHU Summer Workshop 2002]
  - [Duh & Kirchhoff, Coling 2004], [Vergyri, et. al., ICSLP 2004]
- Modeling of conversational speech
  - [Ji & Bilmes, HLT 2004]
- Applied in Speech Recognition, Machine Translation
- General Factored LM tools can also be used to obtain various smoothed conditional probability tables for other applications outside of language modeling (e.g. tagging)
- More possibilities (factors can be anything!)

To explore further...

- Factored Language Model is now part of the standard SRI Language Modeling Toolkit distribution (v.1.4.1)
  - Thanks to Jeff Bilmes (UW) and Andreas Stolcke (SRI)
  - Downloadable at:
    http://www.speech.sri.com/projects/srilm/

fngram Tools

fngram-count -factor-file my.flmspec -text train.txt
fngram -factor-file my.flmspec -ppl test.txt

train.txt: "Factored LM is fun"
W-Factored.P-adj W-LM: P-noun W-is: P-verb W-fun: P-adj

my.flmspec
W: 2 W(-1) P(-1) my.count my.lm 3
W1,P1 W1 kndiscount gtmin 1 interpolate
P1 P1 kndiscount gtmin 1
0 0 kndiscount gtmin 1