Factored Language Models

EE517 Presentation
Kevin Duh (kevinduh@u.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
  \[ p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_1, \ldots, w_{t-1}) \]
  \[ \approx \prod_{t=1}^{T} p(w_t \mid w_{t-1}, w_{t-2}) \]

- How to get robust n-gram estimates \((p(w_t \mid w_{t-1}, w_{t-2}))\)?
  - Smoothing
    - E.g. Kneser-Ney, Good-Turing
  - Class-based language models
    \[ p(w_t \mid w_{t-1}) \approx p(w_t \mid C(w_t))p(C(w_t) \mid C(w_{t-1})) \]
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

<table>
<thead>
<tr>
<th>Arabic k-t-b</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Kitaab</td>
<td>A book</td>
</tr>
<tr>
<td>Kitaab-iy</td>
<td>My book</td>
</tr>
<tr>
<td>Kitaabu-hum</td>
<td>Their book</td>
</tr>
<tr>
<td>Kutub</td>
<td>Books</td>
</tr>
</tbody>
</table>
Arabic Morphology

\( \text{pattern} \)

\( \text{particles} \) fa- sakan -tu \( \text{affixes} \)

\( \text{root} \)

LIVE + past + 1st-sg-past + part: “so I lived”

• \( \sim \)5000 roots
• several hundred patterns
• dozens of affixes
Vocabulary Growth - full word forms

Vocabulary Growth - stemmed words

Solution: Word as Factors

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

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

\[ p(w_1, w_2, \ldots, w_T) \equiv p(f_1^{1:K}, f_2^{1:K}, \ldots, f_T^{1:K}) \]

\[ \approx \prod_{t=1}^{T} p(f_t^{1:K} \mid f_{t-1}^{1:K}, f_{t-2}^{1:K}) \]

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

\[ P(w_t \mid w_{t-1}, w_{t-2}, s_{t-1}, s_{t-2}, m_{t-1}, m_{t-2}) \]
Advantage of Factored Word Representations

- Main advantage: Allows robust estimation of probabilities (i.e. $p(f_t | f_{t-1}^{1:K}, f_{t-2}^{1:K})$) using backoff
  - Word combinations in context may not be observed in training data, but factor combinations are
  - Simultaneous class assignment

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[word]</td>
<td>kitaab-iy</td>
<td>kitaabu-hum</td>
<td>kutub</td>
</tr>
<tr>
<td>[stem]</td>
<td>kitaab</td>
<td>kitaabu</td>
<td>kutub</td>
</tr>
<tr>
<td>[root]</td>
<td>ktb</td>
<td>ktb</td>
<td>ktb</td>
</tr>
<tr>
<td>[tag]</td>
<td>noun+poss</td>
<td>noun+poss</td>
<td>noun (pl.)</td>
</tr>
</tbody>
</table>
Example

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

\[
\begin{align*}
\text{Count}(\text{tiqra, kitAbiy, bi}) &= 0 \\
\text{Count}(\text{tiqra, kutubiy, bi}) &> 0 \\
\text{Count}(\text{tiqra, ktb, bi}) &> 0
\end{align*}
\]

\[P(\text{bi} \mid \text{kitAbiy}, \text{tiqra})\] can back off to
\[P(\text{bi} \mid \text{ktb}, \text{tiqra})\] to obtain more robust estimate.

\[\Rightarrow\] this is better than \[P(\text{bi} \mid <\text{unknown}>, \text{tiqra})\]
Language Model Backoff

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

Word-based LM:

Backoff path: Drop most distant word during backoff

\[
P(W_t | W_{t-1} W_{t-2} W_{t-3}) \rightarrow P(W_t | W_{t-1} W_{t-2}) \rightarrow P(W_t | W_{t-1}) \rightarrow P(W_t)
\]

Factored Language Model:

Backoff graph: multiple backoff paths possible

Factored Language Models
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:**
  \[
P_{BO}(w_t | w_{t-1}, w_{t-2}) = \begin{cases} 
    d_{N(w_t, w_{t-1}, w_{t-2})} \frac{N(w_t, w_{t-1}, w_{t-2})}{N(w_{t-1}, w_{t-2})} & \text{if } N(w_t, w_{t-1}, w_{t-2}) > 0 \\
    \alpha(w_{t-1}, w_{t-2}) P_{BO}(w_t | w_{t-1}) & \text{otherwise}
  \end{cases}
\]

- **Generalized Backoff:**
  \[
P_{BO}(f | f_{P1}, f_{P2}) = \begin{cases} 
    d_{N(f, f_{P1}, f_{P2})} \frac{N(f, f_{P1}, f_{P2})}{N(f_{P1}, f_{P2})} & \text{if } N(f, f_{P1}, f_{P2}) > 0 \\
    \alpha(f_{P1}, f_{P2}) g(f, f_{P1}, f_{P2}) & \text{otherwise}
  \end{cases}
\]

\(g()\) can be any positive function, but some \(g()\) makes backoff weight computation difficult.
g() functions

- A priori fixed path:
  \[ g(f, f_{P1}, f_{P2}) = P_{BO}(f | f_{P1}) \]

- Dynamic path: Max counts:
  \[ g(f, f_{P1}, f_{P2}) = P_{BO}(f | f_{Pj^*}) \]
  \[ j^* = \arg\max_j \quad N(f, f_{Pj}) \]

  Based on raw counts
  => Favors robust estimation

- Dynamic path: Max normalized counts:
  \[ j^* = \arg\max_j \quad \frac{N(f, f_{Pj})}{N(f_{Pj})} \]

  Based on maximum likelihood
  => Favors statistical predictability
Dynamically Choosing Backoff Paths During Training

- Choose backoff path based on $g()$ and statistics of the data
Multiple Backoff Paths: Generalized Parallel Backoff

- Choose **multiple paths** during training and combine probability estimates

\[
p_{bo}(w_t \mid w_{t-1}, s_{t-1}, t_{t-1}) =
\begin{cases}
  d_c p_{ML}(w_t \mid w_{t-1}, s_{t-1}, t_{t-1}) & \text{if count } \geq \text{ threshold} \\
  \frac{\alpha}{2} \left[p_{bo}(w_t \mid w_{t-1}, s_{t-1}) + p_{bo}(w_t \mid w_{t-1}, t_{t-1})\right] & \text{else}
\end{cases}
\]

Options for combination are:
- average, sum, product, geometric mean, weighted mean
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
Example: a Factored LM

- Initial Conditioning Factors, Backoff Graph, and Smoothing parameters completely specify a Factored Language Model.
- E.g. 3 factors total:

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
- Modeling of conversational speech and music
- 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)
References

• Core FLM technology:

• FLM in various applications:
  • Speech recognition / language modeling for morphologically-rich languages
    • Estonian: [T. Alumnae. Sentence-adaptated Factored LM for Transcribing Estonian Speech, ICASSP 2006]

  • Conversational speech modeling
    • G. Ji & J. Bilmes. Multi-Speaker Language Modeling, HLT 2004

  • Machine translation

  • Music
    • X. Li, G. Ji, & J. Bilmes, A Factored Language Model for Quantized Music International Conference on Computer Music (ICMC) 2006
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/

  • Please give me feedback! 😊
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