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

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_1, \ldots, w_{t-1})$$
$$\approx \prod_{t=1}^{T} p(w_t \mid w_{t-1}, w_{t-2})$$

- 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

Arabic k-t-b	
Kitaab	A book
Kitaab-iy	My book
Kitaabu-hum	Their book
Kutub	Books

Arabic Morphology

pattern

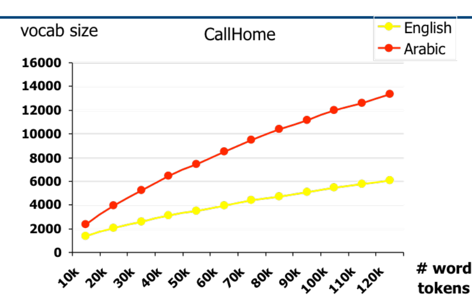
particles fa- sakan -tu *affixes*

root

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

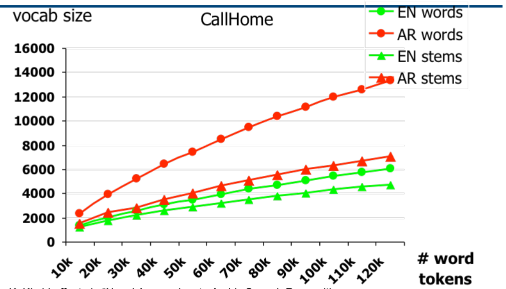
- ~5000 roots
- several hundred patterns
- dozens of affixes

Vocabulary Growth - full word forms



Source: K. Kirchhoff, et al., "Novel Approaches to Arabic Speech Recognition - Final Report from the JHU Summer Workshop 2002", JHU Tech Report 2002

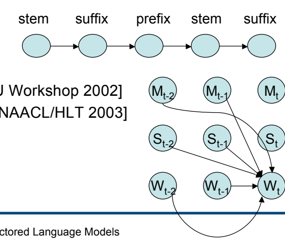
Vocabulary Growth - stemmed words



Source: K. Kirchhoff, et al., "Novel Approaches to Arabic Speech Recognition - Final Report from the JHU Summer Workshop 2002", JHU Tech Report 2002

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]



Factored Word Representations

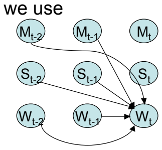
$$w \equiv \{f^1, f^2, \ldots, f^K\} \equiv f^{1:K}$$

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

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

- 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. $p(f_t \mid f_{t-1}^{1:K}, f_{t-2}^{1:K})$) using backoff
  - Word combinations in context may not be observed in training data, but factor combinations are
  - Simultaneous class assignment

Word	Kitaab-iy (My book)	Kitaabu-hum (Their book)	Kutub (Books)
word	kitaab-iy	kitaabu-hum	kutub
stem	kitaab	kitaabu	kutub
root	ktb	ktb	ktb
tag	noun+poss	noun+poss	noun (pl.)

Example

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

Count(tiqra, kitAbiy, bi) = 0
Count(tiqra, kutubiy, bi) > 0
Count(tiqra, ktb, bi) > 0

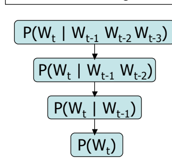
P(bi| kitAbiy, tiqra) can back off to
P(bi I ktb, tiqra) to obtain more robust estimate.
=> this is better than P(bi I <unknown>, tiqra)

Language Model Backoff

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

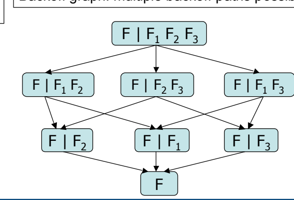
Word-based LM:

Backoff path: Drop most distant word during backoff



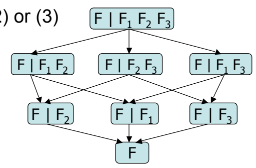
Factored Language Model:

Backoff graph: multiple backoff paths possible



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:

$$P_{BO}(w_t \mid w_{t-1}, w_{t-2}) = \begin{cases} d_{N(w_t, w_{t-1}, w_{t-2})} \dfrac{N(w_t, w_{t-1}, w_{t-2})}{N(w_{t-1}, w_{t-2})} & \text{if } N(w_t, w_{t-1}, w_{t-2}) > 0 \\ \alpha(w_{t-1}, w_{t-2}) P_{BO}(w_t \mid w_{t-1}) & \text{otherwise} \end{cases}$$

- Generalized Backoff:

$$P_{BO}(f \mid f_{P1}, f_{P2}) = \begin{cases} d_{N(f, f_{P1}, f_{P2})} \dfrac{N(f, f_{P1}, f_{P2})}{N(f_{P1}, f_{P2})} & \text{if } N(f, f_{P1}, f_{P2}) > 0 \\ \alpha(f_{P1}, f_{P2}) g(f, f_{P1}, f_{P2}) & \text{otherwise} \end{cases}$$

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

$$\alpha(f_{P1}, f_{P2}) = \frac{1 - \sum_{f : N(f, f_{P1}, f_{P2}) > 0} d_{N(f, f_{P1}, f_{P2})} \dfrac{N(f, f_{P1}, f_{P2})}{N(f_{P1}, f_{P2})}}{\sum_{f : N(f, f_{P1}, f_{P2}) = 0} g(f, f_{P1}, f_{P2})}$$

g() functions

- A priori fixed path:

$$g(f, f_{P1}, f_{P2}) = P_{BO}(f \mid f_{P1})$$

- Dynamic path: Max counts:

$$g(f, f_{P1}, f_{P2}) = P_{BO}(f \mid f_{Pj^*})$$

$$j^* = \operatorname*{argmax}_j N(f, f_{Pj})$$

Based on raw counts
=> Favors robust estimation

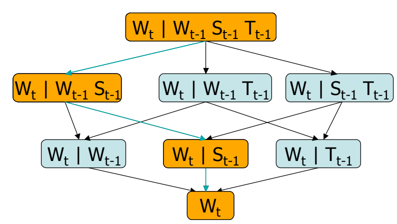
- Dynamic path: Max normalized counts:

$$j^* = \operatorname*{argmax}_j \frac{N(f, f_{Pj})}{N(f_{Pj})}$$

Based on maximum likelihood
=> Favors statistical predictability

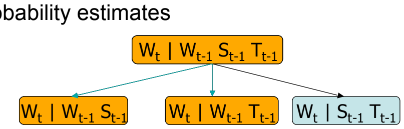
Dynamically Choosing Backoff Paths During Training

- 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



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

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

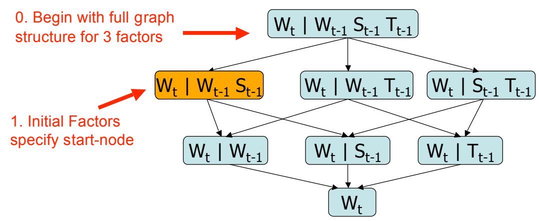
- In n-grams, choose, eg.
  - Bigram vs. trigram vs. 4gram
    => relatively easy search; just try each and note perplexity on development set
- In Factored LM, choose:
  - Initial Conditioning Factors
  - Backoff Graph
  - Smoothing Options

  ⇒Too many options; need automatic search

  ⇒Tradeoff: Factored LM is more general, but harder to select a good model that fits data well.

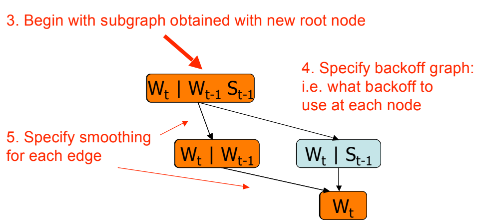
Example: a Factored LM

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



Example: a Factored LM

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



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
```