Language Modeling
- Part 2

June 28, 2012
References

• Review paper from Roni Rosenfeld, IEEE 2000

• Huang et al. Spoken Language Processing, Chapter 11

• Jerome Bellegarda, Speech Communication 2004

Language Modeling - Part 2

• Different Kinds of Language Models
  – Cache Language Models
  – Trigger Language Models
  – Multilevel Language Models
  – Interleaved Language Models
  – Morpheme-Based Language Models
  – Context-Free Grammar Language Models
  – Tree-Based Language Models
  – Hidden Markov Models for Language Modeling

• Practical Issues
  – Spontaneous Speech
  – Unknown Words
  – Different Languages
  – Recognition Errors
  – Combining Language Model and Acoustic Model

• Language Model Adaptation
  – Adaptation Data
  – Adaptation Techniques
Cache Language Models

Observation: When using a speech recognizer (e.g. for dictating news texts) the topics can change at arbitrary points.

- **Idea:**
  
  Use a static and a dynamic component of the language model. Constantly update the dynamic component.

- **Static Component:**

  \[ P_S(w_k | w_{k-(n-1)} \ldots w_{k-1}) \] i.e. the usual trained language model.

- **Dynamic Component:**

  complete n-gram language model constructed from the text dictated so far, e.g.:

  \[ P_{Cache}(w_k | w_{k-(n-1)} \ldots w_{k-1}) = 0.5 \cdot f(w_k) + 0.25 \cdot f(w_k | w_{k-1}) + 0.25 \cdot f(w_k | w_{k-2} w_{k-1}) \]

- **Total Language Model:**

  \[ P_T = \lambda_{Cache} \cdot P_{Cache} + (1- \lambda_{Cache}) \cdot P_S \]

- **Variation:**

  - Compute \( P_{Cache} \) on window of last \( l \) words.
  - Vary \( \lambda_{Cache} \) with the size of the cache
Trigger Language Models

Observation: Often, the probability of a word depends on some words far in the history. Often, when a content-carrying word occurs in a text it is likely to occur again some time later.

• Idea:
  Use a standard interpolated $n$-gram language model.
  But constantly update the unigrams $P(w_k)$ for some words $w_k$.

• Trigger
  For each word $w$ define a trigger list $L(w)$ (possibly weighted) of words that are likely to occur some time later.

E.g. $L($MONEY$) = \{BANK, SAVINGS, ACCOUNT, DOLLARS, COST\}$. Then, for each word $v$ in $L(w)$ increase $P(v)$ by some value.
Multilevel Language Models

• **Idea:**

Use a language model for word sequences $LM_W$, one for phrases $LM_P$, and one for sentences $LM_S$ trained as regular $n$-gram grammars over their corresponding segments of speech.

• **Combination**

Let the total language model be a linear combination of the three independent language models:

$$LM_T = \lambda \cdot LM_W + \lambda \cdot LM_P + \lambda \cdot LM_S$$
Interleaved Language Models

- **Observation:**
  Sentences have many structures. The syntactic constraints are often topic-independent. The sequence of non-content words often is independent from the sequence of content words.

- **Alternation:**

  \[
  \begin{align*}
  \text{LM} & \left( \text{The president of the software company said that they will introduce} \ldots \right) \\
  \text{LM}_{\text{NC}} & \left( \text{The} \quad \text{of the} \quad \text{that they will} \ldots \right) \\
  \text{LM}_{\text{C}} & \left( \text{president} \quad \text{software company said} \quad \text{introduce} \ldots \right)
  \end{align*}
  \]

  \[
  P(w | \text{history}) = \begin{cases} 
  P_C(w | C_1(\text{history})) & \text{if } w \text{ is a content word} \\
  P_{\text{NC}}(w | C_2(\text{history})) & \text{if } w \text{ is a non-content word}
  \end{cases}
  \]
Morpheme-Based Language Models

• Observation:
In heavily inflected (or agglutinating) languages, the word stem is succeeded or preceded by gender-, tempus-, numerus-, and case-identifying suffixes:
Osman-ı-la-r-tırl-ama-yabil-ecek-ler-imiz-den-mi-siniz
(behaving as if you were of those whom we might consider not converting into Ottoman)

The probabilities for the full forms of a word cannot be estimated robustly. However, a robust estimation could be given for the word stem. Also, identifying suffixes depend on the corresponding form of the preceding adjective or article.

• Idea:
Don't compute language model on sequences of words but on sequences of word-building fragments (morphemes):

\[ P(w_1 \, w_2 \, ...) = P(w_{11} \, w_{12} \, w_{13} \, w_{21} \, w_{22} \, ...) \]
Why use context-free grammars (CFG) instead of $n$-grams?

- The rules of the grammar can be written by expert without the need for lots of training text data.
- CFGs are powerful enough to represent large parts of natural languages.
- CFGs are constrained enough to allow efficient search space reduction.
- If interpreted as finite state automaton, then the state transition sequence can also be used for efficient parsing (semantic analysis) of the sentence.
Tree-Based Language Models

Observation:
- $n$-gram models have very limited context
- Parameters are almost un-trainable for large values of $n$

CART-based Technique: (Classification and Regression Trees)
- Each tree node asks question about previous $k$ words ($k$ is approximately 20)
- Each leaf node is a probability distribution on vocabulary (unigrams)
- The tree is automatically generated (including questions)
- Criterion is to minimize leaf node entropy. (preferably clear decisions)
- Compare to clustering of acoustic context-dependent models.

Example:

```
was computer mentioned?
prev word article?
PC 0.1
laptop 0.02
... 
man 0.02
... 
smart 0.2
... 
smart 0.2
... 
fast 0.2
... 
intelligent 0.03
stupid 0.02
```

```
Observation:
- Often, spoken speech is carried out in phases.
  Conversation: greeting phase, small-talk phase, good-bye
  Dictation: Writing a letter: addresses, opening, content, closing
- The typical word sequences in one phase differ from those in another phase.
- We can regard the conversation phases as states of the conversation. Transitions from one state into another can be regarded as a stochastic process.

Approach:
- Build / train HMM that represents conversation phases and transitions, e.g.:

  ![Diagram of HMM states](greeting smalltalk good-bye)

- The states emit language models, or mixture weight distributions (i.e. interpolation factors for different specialized language models).
Summary (Part 2)

• Covered different LM approaches

• Share the same goal: To capture long-range dependence in natural language
  – Cache, Trigger, Tree, HMM, Context Free Grammar, … etc
Practical Issues

- Vocabulary Selection
- N-gram Pruning

- Special Problems with:
  - Spontaneous Speech
  - Unknown Words
  - Different Languages
  - Follow-up Errors
  - Combining Language Model and Acoustic Model
Vocabulary Selection

- **Inflected form** treated as different word in ASR since: different pronunciation, syntactic role, usage, ...

- **Smaller vocabulary** eliminates potential confusable candidates and improves WER

- **BUT:** limiting the vocabulary means
  - severe constraints for the user
  - less flexible system
  - increase Out-of-Vocabulary (OOV) rate

- Problem: Balance out OOV and WER

- Use a corpus of text to determine appropriate vocabularies
  - Use data from **specific topic** or **domain**
  - Pick the **most frequency words**
N-Gram Pruning

- Size of higher-order n-gram models often too large for practical applications
- Reduce the size of the model by **Pruning**
- Prune parameters from n-gram models:
  - minimize relative entropy between original and pruned model
  - minimize performance loss
- Choose n-grams as to:
  - maximize performance (i.e. minimize perplexity)
- Experience: Trigram models can be pruned by more than 25% without degrading recognition performance (Stolcke)
Special Problems with Spontaneous Speech

In spontaneous speech, we often observe words that blow up the word sequence but in most cases don't influence the language model:

- Silence between words can be optionally inserted or omitted
- Hesitations: AH, UHs, UMs, ...
- Filler words and phrases: YEAH, YOU KNOW, ...
- False starts, aborted or stuttered words: LET’S MEET SAT- NO SUNDAY
- Non verbal sounds: breathing, lip smacks, tongue clicks, ...

Such words are narrowing the context of $n$-grams, sometimes even push out the entire content-carrying context.

Approaches:
- Ignore any problem, treat spontaneous effects like regular words
- Increase context width (use four-grams, etc.)
- Skip spontaneous effects in the history of $P(w \mid \text{history})$
- Model only those events which do have an impact and skip the other (e.g. breathing at semantic unit boundaries)
Special Problems with Unknown Words

Observation:
• In acoustic modeling, we can handle words that have not been seen in the training set by simply defining their pronunciation in the pronunciation lexicon.
• In most speech recognition tasks, we can not train all words that might have to be recognized some time.

Question:
How do we incorporate unknown words into the language model
BTW: here unknown word refers to a word not seen in the training but occurring in the decoder vocabulary and dictionary – as opposed to be not known at all

Approaches:
• Rewrite all words in the training text that occur only once with the word "UNK", treat every unknown word in the recognition run as if it was "UNK".
• Use a class-based language model, assign a class to every unknown word and use the class probabilities.
• Optionally add new word into vocabulary and use cache language model to improve the parameters of frequently occurring new words.
**Special Problems with OOV Words**

**OOV (out-of-vocabulary) words:**
Words that neither occur in training NOR in test vocabulary

**OOV Rate:**
Relative frequency of OOV words in the test set

**Recognition Errors:**
OOV word MUST lead to errors in recognition;
Best case scenario:
1 OOV word – 1 substitution
i.e. WER=OOV-rate

**BUT: Follow-up errors**
- Observation: Every OOV word causes between 1.5 and 2.0 errors in average.

<table>
<thead>
<tr>
<th>Language</th>
<th>Vocabulary</th>
<th>OOV-Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Korean</td>
<td>64K</td>
<td>34.0%</td>
</tr>
<tr>
<td>Turkish</td>
<td>64K</td>
<td>13.5%</td>
</tr>
<tr>
<td>German</td>
<td>61K</td>
<td>4.4%</td>
</tr>
<tr>
<td>Portuguese</td>
<td>60K</td>
<td>4.3%</td>
</tr>
<tr>
<td>English</td>
<td>64K</td>
<td>0.3%</td>
</tr>
<tr>
<td>Korean (segmented)</td>
<td>64K</td>
<td>0.2%</td>
</tr>
<tr>
<td>Chinese (segmented)</td>
<td>60K</td>
<td>0%</td>
</tr>
<tr>
<td>Croatian</td>
<td>31K</td>
<td>13.6%</td>
</tr>
<tr>
<td>Spanish</td>
<td>30K</td>
<td>5.2%</td>
</tr>
<tr>
<td>French</td>
<td>30K</td>
<td>4.7%</td>
</tr>
<tr>
<td>Japanese (segmented)</td>
<td>22K</td>
<td>3.0%</td>
</tr>
</tbody>
</table>

Table 1.2: Out-Of-Vocabulary Rates for 10 Languages
Vocabulary Growth for Different Languages and Domains

![Graph showing vocabulary growth for different languages and domains.](image-url)
Languages differ in their degree of inflection and definition of words

- Highly inflecting languages pose more problems to language modeling (e.g. Turkish, Korean)
  => Use **morpheme-based LMs**, and syntactic/semantic classes

- Languages differ in their potential to form new words.
  (e.g. German allows arbitrary compounding of nouns)
  => **Decompose compound nouns** for calculating the language model

- Some languages have a different or no specified notion of words
  (e.g. Chinese 曲阜孔子博物馆)
  Where does a word start and where does it end?
  => Consider **syllable based language models**
Follow-up errors due to strong language model constraints:

- If a word is misrecognized, a strong language model constraint can force the next word(s) to be misrecognized, too.

- E.g. spoken: "The dog barks."
  If "dog" is misrecognized as "fog", the probability for "barks" becomes very small.
  => no really working cure.

Approaches:
Change the effect of the language model on the recognition process depending on the current confidence.
Combining Language Model and Acoustic Model

The fundamental problem of speech recognition is:

\[ W' = \arg\max_W P(W \mid X) = \frac{p(X \mid W) \cdot P(W)}{p(X)} = \arg\max_W p(X \mid W) \cdot P(W) \]

\( W \Rightarrow \) many Most language models do not take into account the probability of the length of a word sequence insertion or deletion errors.

Remedy: \( \arg\max_W p(X \mid W) \cdot P(W) \cdot q^{\mid W} \)

While \( P(W) \) is a probability, \( p(A \mid W) \) is a density \( \Rightarrow \) variance problems when combining them.

Remedy: \( \arg\max_W p(X \mid W) \cdot P(W)^z \cdot q^{\mid W} \)

The parameters \( z \) and \( q \) must be optimized on a cross-validation set.
Language Model Adaptation
Jerome Bellegarda, Apple Computer, Speech Communication 2003 – best paper award

• Why adaptation?
• Natural language is highly variable in several aspects:
  – Language evolves, vocabulary changes with time
    (try to read a VERY old book)
  – Different domains have different word sequences
    (“interest rate” in banking application vs. gaming)
  – People adjust language use based on task
    (compare your scientific paper to your email to a friend)
  – Discourse style varies due to socio-economic status, emotion, …
• Lexicon, syntactic and semantic characteristics of training and recognition tasks differ!
• Bad news for n-gram modeling as mismatch always hurts!
  – Example: phone conversation - 2 Mio transcripts better than 140 Mio of BN
  – Example: Dow-Jones newswire doubles pp when applied to AP newswire
  – Linguistic mismatch affects recognition accuracy way more than acoustic!!
Adaptation Framework

**Corpus A**: a (small) adaptation corpus, relevant to the current task
**Corpus B**: a (large) background corpus, somewhat different task
**Goal**: compute a suitable robust estimate of the LM probabilities

![Adaptation Diagram]

Fig. 1. General framework for SLM adaptation.
Adaptation Problem

- Goal: Compute a robust estimate of the LM probability

\[ \Pr(w_1, \ldots, w_N) = \prod_{q=1}^{N} \Pr(w_q|h_q), \]

- Where \( h_q \) represents the history available at time \( q \)
- For an n-gram we have

\[ h_q = w_{q-n+1}, \ldots, w_{q-1} \]

- So, the estimation of \( \Pr (w_1, w_2, \ldots w_N) \) leverages two distinct knowledge sources:
  - The well-trained but possibly mismatched **background SLM**
  - **Adaptation data** which is used to extract specific information relevant to the current task
Techniques for LM Adaptation

- Idea: Dynamically modify background SLM estimate on the basis of what can be extracted from corpus A (relevant to the current task).

- LM Adaptation Techniques fall into three broad areas:
  1. Model Interpolation:
     Interpolate dynamic (task-specific) with static (background) SLM
  2. Constraint Specification:
     Extract features from corpus A that have to be met by the adapted SLM
  3. Meta-Information Extraction:
     Include knowledge that is not explicitly observable in the word sequence itself
     - Underlying discourse topic
     - General semantic and syntactic information
     - Combination of both
Adaptation Data

- **How to get Adaptation Data A?**
  - Might be available from existing system build, e.g.
    - Wizard of Oz experiments for dialog systems
  - If not available OR corpus A is too small, we need to collect data

1. **Grammar-based system**: Generate text using the grammar. Choose rules randomly for creating an artificial corpus (e.g. Monte Carlo) or use small corpus to learn rule weights

2. **Use recognition output**: Accumulate corpus A during the recognition process itself; Take N-best list or Confusion Networks as adaptation material; each hypothesis contribute to corpus according to its posterior likelihood (well-recognized parts occur in most hypos, while misrecognized are not enforced)

3. **Search & Retrieve**: Use small corpus A as seed for retrieval techniques (online databases and world wide web)

4. Augment data at the n-gram count level
Method 1: Model Interpolation

Idea: Derive task-specific (dynamic) SLM from Corpus A and combine this with the background (static) SLM from B

A. Model Merging: Linear Interpolation
Simplest form is linear interpolation, where interpolation coefficient can be a function of the word history or word class history

\[
\Pr(w_q|h_q) = (1 - \lambda) \Pr_A(w_q|h_q) + \lambda \Pr_B(w_q|h_q)
\]

Estimate \( \lambda \) on held-out data (subset of A)
Do it empirically
Use EM algorithm to maximize likelihood
If no held-out data: do leaving-one-out
Method 1: Model Interpolation

Idea: Derive task-specific (*dynamic*) SLM from Corpus A and combine this with the background (*static*) SLM from B.

A. Model Merging: Back-off

Fill-up technique: Back-off from dynamic to static estimate depending on the frequency count.

\[
\Pr(w_q|h_q) = \begin{cases} 
\Pr_A(w_q|h_q) & \text{if } C_A(h_q w_q) \geq T; \\
\beta \Pr_B(w_q|h_q) & \text{otherwise},
\end{cases}
\]

B. Dynamic Cache Model

C. MAP adaptation
Method 1: Model Interpolation

Idea: Derive task-specific (dynamic) SLM from Corpus A and combine this with the background (static) SLM from B

B. Dynamic Cache Model

Special case of linear interpolation: self-trigger words inside corpus A to capture short-term dynamic shifts in word-use frequencies

\[
Pr(w_q | h_q) = \sum_{\{c_q\}} Pr(w_q | c_q) Pr(c_q | h_q),
\]

- set of possible classes for \( w_q \) given history \( h_q \)
- class assignment component subject to dynamic cache adaptation
- class \( n \)-gram component assumed to be task independent, thus taken from corpus B

\[
Pr(w_q | c_q) = (1 - \lambda)Pr_A(w_q | c_q) + \lambda Pr_B(w_q | c_q)
\]
Method 1: Model Interpolation

Idea: Derive task-specific (dynamic) SLM from Corpus A and combine this with the background (static) SLM from B

C. MAP adaptation (Maximum A Posteriori)

Combine at the frequency count level rather than at the model level.

$\varepsilon$ is a constant factor that is estimated empirically, can be a function of the word history and used for tuning towards domain-specific properties

$$
Pr(w_q|h_q) = \frac{\varepsilon C_A(h_q w_q) + C_B(h_q w_q)}{\varepsilon C_A(h_q) + C_B(h_q)}
$$
Method 2: Constraint Specification

Idea: Extract features from corpus A that are used as constraints for the adapted SLM

More powerful than Model Interpolation since different weights could be assigned to each extracted feature separately

A. Exponential Models
Historically constraint-based methods are associated with exponential models trained on Maximum Entropy (ME) criterion which leads to Minimum Discrimination Information (MDI) estimation

B. MDI Adaptation

C. Unigram Constraints
Method 3: Topic Information

Idea: Use corpus A to extract information about the underlying subject matter. Improve upon the background model based on semantic classification.

Mixture Models
Semantic Knowledge
Syntactic Infrastructure
  - Structured Language Models
  - Syntactic Triggers
Multiple Sources
  - Combination Models
  - Whole sentence Models
Language Modeling Summary

- Deterministic vs. Stochastic Language Modeling
- Classification of Word Sequence Histories
- N-grams
- The Perplexity of Language Models (quality measure)
- N-gram Smoothing (“Collect tax from the rich, redistribute it to the poor”)
- Different Kinds of Language Models
  - Cache, Trigger, Multilevel, Interleaved, Morpheme-Based, Context-Free Grammar, Tree-Based, HMM-based Language Models
- Practical Issues
- Vocabulary Selection
- Spontaneous Speech
- Unknown Words
- Different Languages
- Out-of-Vocabulary Words