Acknowledgments: Some slides were adapted from Victor Lavrenko (Edinburgh), Cheng Zhai (UIUC), and Don Metzler (Google)
Language Models: Intuition

Metzler & Lavrenko, SIGIR 2009 Tutorial

\[
\log P(Q|D) = \sum_{w \in Q \cap D} \log \left( 1 + \frac{\lambda_D}{1 - \lambda_D} \frac{tf_{w,D}}{|D|} \frac{|C|}{cf_w} \right)
\]

Repetitions of query words → good

Common words less important

More words in common with the query → good

Repetitions less important than different query words

\[
\log (1 + tf_w)
\]

Copyright Don Metzler, Victor Lavrenko
LMs are Generative ~ Explanatory

• How do we determine if a given model is a LM?
  • LM is generative
    – at some level, a language model can be used to generate text
    – explicitly computes probability of observing a string of text
    – Ex: probability of observing a query string from a document model
      probability of observing an answer from a question model
    – model an entire population

• Discriminative approaches
  – model just the decision boundary
  – Ex: is this document relevant?
    does it belong to class X or Y

  – have a lot of advantages,
  - but these are not generative approaches
LMs for IR: Recap

- Use Unigram models
  - no consistent benefit from using higher order models
  - estimation is much more complex (e.g. bi-gram from a 3-word query)

- Use Multinomial models
  - well-studied, consistent with other fields that use LMs
  - extend multiple-Bernoulli model to non-binary events?

- Use Model Comparison for ranking
  - allows feedback, expansion, etc. through estimation of MQ and MD
  - use $KL(MQ \mid\mid MD)$ for ranking multiple documents against a query

- Estimation of MQ and MD is a crucial step
  - very significant impact on performance (more than other choices)
  - key to cross-language, cross-media and other applications
More Advanced LMs: +Feedback

Classic Prob. Model

\[ O(R = 1 | Q, D) \propto \frac{P(D | Q, R = 1)}{P(D | Q, R = 0)} \]

Rel. doc model

NonRel. doc model

Query likelihood ("Language Model")

\[ O(R = 1 | Q, D) \propto P(Q | D, R = 1) \]

"Rel. query" model

Initial retrieval:
- query as rel doc vs. doc as rel query
- \( P(Q|D,R=1) \) is more accurate

Feedback:
- \( P(D|Q,R=1) \) can be improved for the current query and future doc
- \( P(Q|D,R=1) \) can also be improved, but for current doc and future query

Parameter Estimation

\[
\begin{align*}
(q_1,d_1,1) & \quad \quad P(D|Q,R=1) \\
(q_1,d_2,1) & \quad \quad P(D|Q,R=1) \\
(q_1,d_3,1) & \quad \quad P(D|Q,R=0) \\
(q_1,d_4,0) & \quad \quad P(D|Q,R=0) \\
(q_1,d_5,0) & \quad \quad P(D|Q,R=0) \\
(q_3,d_1,1) & \quad \quad P(Q|D,R=1) \\
(q_4,d_1,1) & \quad \quad P(Q|D,R=1) \\
(q_5,d_1,1) & \quad \quad P(Q|D,R=1) \\
(q_6,d_2,1) & \quad \quad P(Q|D,R=1) \\
(q_6,d_3,0) & \quad \quad P(Q|D,R=1)
\end{align*}
\]

Query-based feedback
Doc-based feedback
Relevance Model Estimation

- Question: How to estimate $P(D|Q,R)$ (or $p(w|Q,R)$) without relevant documents?

- Key idea:
  - Treat query as observations about $p(w|Q,R)$
  - Approximate the model space with document models

- Two methods for decomposing $p(w,Q)$
  - **Independent sampling (Bayesian model averaging)**
    
    $p(w|Q,R) = \int p(w|\theta_D)p(\theta_D|Q,R)d\theta_D \approx \int p(w|\theta_D)p(\theta_D|R)p(Q|\theta_D)d\theta_D$

    $\approx \sum_{D \in C} p(w|\theta_D)p(\theta_D|R)p(Q|\theta_D) \propto \sum_{D \in C} p(w|\theta_D) \prod_{j=1}^{m} p(q_j|\theta_D)$

  - **Conditional sampling:** $p(w,Q) = p(w)p(Q|w)$

    $p(w|Q,R = 1) \propto p(w)p(Q|w) = p(w)\prod_{i=1}^{m} \sum_{D \in C} p(q_i|D)p(D|w)$

    $p(w) = \sum_{D \in C} p(w|D)p(D)$

    $p(D|w) = \frac{p(w|D)p(D)}{p(w)}$ 

    Original formula in [Lavranko & Croft 01]
Query Model Estimation

• Question: How to estimate a better query model than the ML estimate based on the original query?

• “Massive feedback”: Improve a query model through co-occurrence patterns learned from
  – A document-term Markov chain that outputs the query [Lafferty & Zhai 01b]
  – Thesauri, corpus [Bai et al. 05, Collins-Thompson & Callan 05]

• Model-based feedback: Improve the estimate of query model by exploiting pseudo-relevance feedback
  – Update the query model by interpolating the original query model with a learned feedback model [Zhai & Lafferty 01b]
  – Estimate a more integrated mixture model using pseudo-feedback documents [Tao & Zhai 06]
Feedback as Model Interpolation

Document D $\rightarrow \theta_D$

Query Q $\rightarrow \theta_Q$

$D(\theta_Q \parallel \theta_D) \rightarrow$ Results

$\theta_Q' = (1 - \alpha)\theta_Q + \alpha\theta_F$

$\alpha = 0 \quad \alpha = 1$

$\theta_Q' = \theta_Q \quad \theta_Q' = \theta_F$

No feedback Full feedback

Feedback Docs $F = \{d_1, d_2, \ldots, d_n\}$

Generative model

Divergence minimization
Estimation Method: Mixture Model

\[
\log p(F \mid \theta) = \sum \prod_{D \in F, w \in D} c(w; D) \log((1 - \lambda)p(w \mid \theta) + \lambda p(w \mid C))
\]

Maximum Likelihood \[\theta_F = \arg \max_{\theta} \log p(F \mid \theta)\]
Example: Query Model with Feedback

Trec topic 412: “airport security”

<table>
<thead>
<tr>
<th>λ = 0.9</th>
<th>Web database</th>
<th>Top 10 docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>security</td>
<td>0.0558</td>
<td>the</td>
</tr>
<tr>
<td>airport</td>
<td>0.0546</td>
<td>security</td>
</tr>
<tr>
<td>beverage</td>
<td>0.0488</td>
<td>airport</td>
</tr>
<tr>
<td>alcohol</td>
<td>0.0474</td>
<td>beverage</td>
</tr>
<tr>
<td>bomb</td>
<td>0.0236</td>
<td>alcohol</td>
</tr>
<tr>
<td>terrorist</td>
<td>0.0217</td>
<td>to</td>
</tr>
<tr>
<td>author</td>
<td>0.0206</td>
<td>of</td>
</tr>
<tr>
<td>license</td>
<td>0.0188</td>
<td>and</td>
</tr>
<tr>
<td>bond</td>
<td>0.0186</td>
<td>author</td>
</tr>
<tr>
<td>counter-terror</td>
<td>0.0173</td>
<td>bond</td>
</tr>
<tr>
<td>terror</td>
<td>0.0142</td>
<td>bomb</td>
</tr>
<tr>
<td>newsnet</td>
<td>0.0129</td>
<td>terrorist</td>
</tr>
<tr>
<td>attack</td>
<td>0.0124</td>
<td>in</td>
</tr>
<tr>
<td>operation</td>
<td>0.0121</td>
<td>license</td>
</tr>
<tr>
<td>headline</td>
<td>0.0121</td>
<td>state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>λ = 0.7</th>
<th>Web database</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.0405</td>
</tr>
<tr>
<td>security</td>
<td>0.0377</td>
</tr>
<tr>
<td>airport</td>
<td>0.0342</td>
</tr>
<tr>
<td>beverage</td>
<td>0.0305</td>
</tr>
<tr>
<td>alcohol</td>
<td>0.0304</td>
</tr>
<tr>
<td>to</td>
<td>0.0268</td>
</tr>
<tr>
<td>of</td>
<td>0.0241</td>
</tr>
<tr>
<td>and</td>
<td>0.0214</td>
</tr>
<tr>
<td>author</td>
<td>0.0156</td>
</tr>
<tr>
<td>bomb</td>
<td>0.0150</td>
</tr>
<tr>
<td>terrorist</td>
<td>0.0137</td>
</tr>
<tr>
<td>in</td>
<td>0.0135</td>
</tr>
<tr>
<td>license</td>
<td>0.0127</td>
</tr>
<tr>
<td>state</td>
<td>0.0127</td>
</tr>
<tr>
<td>by</td>
<td>0.0125</td>
</tr>
</tbody>
</table>
Alternative approach: Translation

- Basic LMs do not address issues of synonymy.
  - Or any deviation in expression of information need from language of documents

- A translation model lets you generate query words not in document via “translation” to synonyms etc.
  - Or to do cross-language IR, or multimedia IR

\[
P(\tilde{q} \mid M) = \prod_i \sum_{v \in \text{Lexicon}} P(v \mid M) T(q_i \mid v)
\]

  Basic LM  Translation

  - Need to learn a translation model (using a dictionary or via statistical machine translation)
Relaxing Independence Assumption

Dependence Language Model for Information Retrieval, SIGIR 2004, Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, Guihong Cao

• Assume: query terms are *linked*:

  ![Diagram of linked terms]

  (how)(has)affirmative action affected (the) construction industry

• Two-step generative process of \( Q \mid D \):

\[
P(Q \mid D) = \sum_{L} P(Q, L \mid D) = \sum_{L} P(L \mid D)P(Q \mid L, D)
\]

• Estimating \( P(L \mid D) \): use statistical dependency parsing:

\[
P(L \mid Q) = \prod_{l \in L} P(l \mid Q)
\]
LMs with Dependency Linkage (2)

- Estimating $P(q_i|D)$

\[
P'(q_i | D) = (1 - \lambda)P(q_i | D) + \lambda P(q_i | C)
\]

\[
= (1 - \lambda) \frac{C_D(q_i) + C_C(q_i)}{\sum q_i C_C(q_i) + \mu} + \lambda \frac{C_C(q_i) - \delta}{\sum q_i C_C(q_i)}
\]

- Results (TREC, WSJ, FR, AP):

<table>
<thead>
<tr>
<th>Models</th>
<th>AvgP</th>
<th>% change over BM</th>
<th>% change over UG</th>
<th># of dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>18.62</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>UG</td>
<td>18.28</td>
<td>-1.83</td>
<td>--</td>
<td>7.2E5 unigram</td>
</tr>
<tr>
<td>CM</td>
<td>18.53</td>
<td>-0.48</td>
<td>+1.4</td>
<td>5.2E7</td>
</tr>
<tr>
<td>DM</td>
<td>19.64</td>
<td>+5.48*</td>
<td>+7.4*</td>
<td>2.5E7</td>
</tr>
</tbody>
</table>
Some Applications of SLMs

• Query performance prediction/ambiguity
• Cross-language Retrieval
• Image Retrieval
Estimating Query Difficulty: Ambiguity

American Airlines or Alcoholics Anonymous?
Query Clarity

- Clarity score ~ low ambiguity
- Cronen-Townsend et. al. SIGIR 2002
- Compare a language model
  - over the relevant documents for a query
  - over all possible documents

- The more difference these are, the more clear the query is
- “programming perl” vs. “the”
Clarity score

\[
\text{Clarity score} = \sum_{w \in V} P(w | Q) \log_2 \frac{P(w | Q)}{P_{\text{coll}}(w)}
\]
Predicting Query Difficulty
[Cronen-Townsend et al. 02]

• Observations:
  – Discriminative queries tend to be easier
  – Comparison of the query model and the collection model can indicate how discriminative a query is

• Method:
  – Define “query clarity” as the KL-divergence between an estimated query model or relevance model and the collection LM

\[
\text{clarity}(Q) = \sum_w p(w | \theta_Q) \log \frac{p(w | \theta_Q)}{p(w | Collection)}
\]

  – An enriched query LM can be estimated by exploiting pseudo feedback (e.g., relevance model)

• Correlation between the clarity scores and retrieval performance
Clarity scores on TREC-7 collection

- Train: 0.33 (0.51)
  - Train dog: 0.65 (0.66)
  - Obedience train dog: 2.43 (2.73)
- Railroad train: 0.73 (0.91)
  - Railroad train dog: 0.67 (0.43)
  - Railroad train caboose: 1.46 (1.24)
Can use many more features

- http://www.slideshare.net/DavidCarmel/sigir12-tutorial-query-perfromance-prediction-for-ir
Cross-Lingual Retrieval
Metzler & Lavrenko, SIGIR 2009 Tutorial

Diagram:
- User submits a query.
- Query is translated.
- Translated query is sent to search engine(s) for other languages.
- Retrieved documents in other languages are returned.
- Translated query is also translated again for another round of retrieval.
CLIR: Query Translation Approach

- Translating documents usually infeasible
- Automatic translation: ambiguous process
  - query as a whole: usually not a well-formed utterance
  - word-for-word: multiple candidate translations
    - environment → environnement, milieu, atmosphere, cadre, conditions
    - protection → garde, protection, preservation, defense, racket
    - agency → agence, action, organisme, bureau
- How to combine translations?
  - single bag of words: bias to multi-meaning words
  - combinations / hypotheses
    - How many? How to assign weights?
Translation model: set of probabilities $P(e|f)$
- probability that French word "f" translates to English word "e"
  - e.g. $P("environment" | "milieu") = \frac{1}{4}$, $P("agency" | "agence") = \frac{1}{2}$, etc.

Language model of a French document: $P(f|M_D)$
- probability of observing "f": $P(milieu|M_D) = \frac{f_{milieu,D}}{|D|}$

Combine into noisy-channel model:
- author writes a French document by sampling words from $M_D$
- channel garbles French words into English according to $P(e|f)$
- probability of receiving an English word: $P(e|M_D) = \sum_f P(e|f)P(f|M_D)$
Language Models for CLIR (cont’d)

Metzler & Lavrenko, SIGIR 2009 Tutorial

- How to estimate $P(e|f)$?
- $f \rightarrow e$ dictionary: assign equal likelihoods to all translations
  - agence → agency: 1/5, bureau: 1/5, branch: 1/5, office: 1/5, service: 1/5
- $e \rightarrow f$ dictionary: use Bayes rule, collection frequency
  - agency → agence: ¼, action: ¼, organisme: ¼, bureau: ¼
  - $P(\text{agency}|\text{agence}) = P(\text{agence}|\text{agency}) \times P(\text{agency}) / P(\text{agence})$
- parallel corpus:
  - set of parallel sentences \{E,F\} such that E is a translation of F
  - simple co-occurrence: how many times e,f co-occur: $P(e|f) = \frac{|\{(E,A): e \in E \land f \in F\}|}{|F: f \in F|}$
  - IBM translation model 1:
    - alignment: links between English, French words
    - count how many times e,f are aligned
      - clean your coffee cup
      - nettoyer votre tasse de café
LMs for CLIR: Putting it all together

Metzler & Lavrenko, SIGIR 2009 Tutorial

- Rank documents by
  \[
P(e_1 \ldots e_k | M_D) = \prod_{i=1}^{k} (\lambda D P(e | M_D) + (1 - \lambda D) P(e))
\]
  - probability English query generated from French document
  - formal, effective model (75-95% of monolingual IR)
  - query expansion: multiple French words translate to "agency"

- Important issues:
  - translation probabilities ignore context
    - one solution: treat phrases as units, but there's a better way
  - vocabulary coverage extremely important
  - morphological analysis crucial for Arabic, Slavic, etc.
  - no coverage for proper names → transliterate:
    - Qadafi, Kaddafi, Qathafi, Gadafi, Qaddafy, Quadhaffi, al-Qaddafi, ..
CLIR: Tricks: Triangulated Translation

Metzler & Lavrenko, SIGIR 2009 Tutorial

• Translation models need bilingual resources
  - dictionaries / parallel corpora
  - not available for every language pair (Bulgarian → Tagalog)

• Idea: use resource-rich languages as interlingua:
  - map Tagalog → Spanish, then Spanish → Bulgarian
  - use multiple intermediate languages, assign weights

• Results slightly exceed direct bilingual resource
Image Retrieval

• Task: Find images related to given query
• Need: association between words and images

{ tiger, grass, trees }
Language Modeling Approach

- Query is a bag of words: \{tiger, grass, trees\}
- Convert every image to a bag of word-like units
- Reduces to cross-language retrieval problem
  - given a query in English: “tiger grass”
  - match documents written in foreign (visual) “words”
- Main issues:
  - how do we define / compute these visual words?
  - is the cross-language retrieval model sufficient?
From Images to Words

- Convert into a set of discrete “features”
  - break image into a set of patches
    - “grassy”, “watery”, “tigery” patches
    - captures different objects in image
  - extract features for each patch
    - reflect visual appearance of a patch
    - relative position, color histogram, texture filters
  - replace feature vector with a discrete label
    - meaningful label (e.g. “grass”) needs human annotations
    - clustering: group feature vector with other, similar vectors
Clustering to group similar vectors

Use clustering (e.g. K-means) to group similar feature vectors from every patch of every image we have.

A cluster label represents a group of similar-looking patches (across all images in the dataset).
Cluster IDs as “Words”

- After clustering:
  - every patch of every image falls into some cluster
    - all similar-looking patches fall into the same cluster
    - cluster id says something about patches that fall into it
      - “C27” → green, vertically-textured
  - Use cluster ids as “words”
    - \[ D = \{ 4 \times \text{“C14”}, 7 \times \text{“C27”}, 24 \times \text{“C79”}, 0 \times \text{everything else} \} \]
    - similar to controlled vocabulary / category codes
      - discrete, content-bearing, Zipfian distribution
      - sometimes called “vis-terms” or “visual words”
Retrieving with LMs

- Converted:
  \{ 4 \times "C14", 7 \times "C27", 24 \times "C79" \}

- Want to query with "tiger", not "C14"

  - use LMs to “translate” English queries into vis-terms
  - rank images by probability they “generate” query:

    \[
    P(e_1...k|I) = \prod_{i=1}^{k} \left( \lambda_i \sum_{v} P(e_i|v) P(v|I) + (1 - \lambda_i) P(e_i) \right)
    \]

    - translate to English
    - draw a visterm
    - Smoothing (IDF)
    - probability that one of the visterms present in the image “translates” to query word \( e_i \)

- need two components:
  - \( P(v|I) \) … document model based on counts of vis-terms
  - \( P(q|v) \) … model for associating words \( q \) with visterms \( v \)
Estimating Translation Probabilities

- No dictionaries
- Parallel corpora (manually-tagged images)
  - e.g.: Corel, Pascal VOC, TREC Vid, LabelMe
  - pre-process → get paired sets: \( \{v_1 \ldots v_n, e_1 \ldots e_m\} \)
- extract translation pairs \( P(e|v) \)
  - co-occurrence model (direct count): \( P(e|v) = \frac{|I:e \in E_I, v \in V_I|}{|I:v \in V_I|} \)
    - problem: will associate “tiger” with C14 and C27 and C79
  - IBM translation model 1
    - uses EM to align “tiger” → C14, “grass” → C27, etc.
- problem: visterms don't map to words 1-1
  - in isolation does not “translate” to anything
Set-based Translation

- Visterms $\leftrightarrow$ tags is a set-to-set mapping
  - don't try to break it into pairs: model holistically
  - joint probability of a set of tags w. a set of visterms
    - cross-media relevance model:

$$P(e_1\ldots e_m, v_1\ldots v_n) = \sum_{E,V} \prod_i P(e_i|E) \cdot \prod_j P(v_j|V) \cdot P(E,V)$$

- note: can't just count: $\{e_1\ldots e_m, v_1\ldots v_n\}$

- Annotate with set of tags: $\arg\max_{e_1\ldots e_m} P(e_1\ldots e_m, v_1\ldots v_n)$

- Rank images by: $P(e_1\ldots e_m|v_1\ldots v_n) = \frac{P(e_1\ldots e_m, v_1\ldots v_n)}{P(v_1\ldots v_n)}$
Topic (aspect) Language Models: Preview

Information need

\[
P(Q | M_C, M_T) \quad \text{and} \quad P(Q | M_C, M_T, M_d)
\]

generation

query

document collection

\[\begin{align*}
M_C & \quad M_{T1} & \quad M_{d1} & \quad d1 \\
M_{T2} & \quad M_{d2} & \quad d2 \\
\vdots & \quad \vdots & \quad \vdots \\
M_{Tm} & \quad M_{dn} & \quad dn
\end{align*}\]
Researchers at the University of Massachusetts biology department announced a breakthrough in the genetic sequencing of...

The exact nature of black hole formation has been mystifying physicists for the past...

- Astronomy
+ Microbiology
- Genetics
+ Physics

Astronomy
- universe
- galaxies
- clusters
- matter
- galaxy
- cluster
- cosmic
- dark
- light
- density

Microbiology
- bacteria
- bacterial
- resistance
- coli
- strains
- microbial
- strain
- salmonella
- resistant

Genetics
- sequence
- sequences
- genome
- dna
- sequencing
- map
- genes
- chromosome
- regions
- human

Physics
- theory
- physics
- physicists
- einstein
- university
- gravity
- black
- theories
- aps
- matter

Corpus