Statistical Language Modeling for Information Access

Theory III: More models, more knowledge

Maarten de Rijke Edgar Meij Krisztian Balog

University of Amsterdam
Norwegian University of Science and Technology

August 1–4, 2011

Outline

Outline

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Improving on Basic LMs

- Capturing limited dependencies
  - Bigrams, trigrams (Song and Croft 1999; see yesterday's lecture), Grammatical dependency (Nallapati and Allan, 2002; Gao et al, 2005; see later today)
  - Generally insignificant improvement as compared to extensions such as feedback
- Full Bayesian query likelihood (Zaragoza et al, 2003)
  - Performance similar to basic LM approach
- Translation model for $p(q|d, R)$
  - Address polysemy and synonyms; improves over basic LMs but expensive; see next slide
- Cluster-based smoothing/scoring
  - Improves over basic LM, but expensive; see slide after next
- Relevance models: principled way of brining in feedback; later today
- Parsimonious LMs: mixture model to filter out non-discriminative words

Translation Models

- No this is not about cross-lingual IR…
- Directly modeling the "translation" relationship beteen words in query and words in doc

$$p(q|d) = \prod_i \sum_{w_j \in V} p_t(t_i|w_j)p(w_j|d)$$

$p_t(t_i|w_j)$: translation model
$p(w_j|d)$: regular document LM

- When relevance judgments are available, (q, d) pairs serve as data to train translation model
- Without relevance judgments, use synthetic data, <title, body> pairs, thesauri

Translation Models

- No this is not about cross-lingual IR...
- Directly modeling the "translation" relationship beteen words in query and words in doc

$$p(q|d) = \prod_i \sum_{w_j \in V} p_t(t_i|w_j) p(w_j|d)$$

  $p_t(t_i|w_j)$: translation model
  $p(w_j|d)$: regular document LM
- When relevance judgments are available, (q, d) pairs serve as data to train translation model
- Without relevance judgments, use synthetic data, <title, body> pairs, thesauri

Translation Models

- No this is not about cross-lingual IR...
- Directly modeling the "translation" relationship beteen words in query and words in doc

$$p(q|d) = \prod_i \sum_{w_j \in V} p_t(t_i|w_j)p(w_j|d)$$

  $p_t(t_i|w_j)$: translation model
  $p(w_j|d)$: regular document LM

- When relevance judgments are available, (q, d) pairs serve as data to train translation model
- Without relevance judgments, use synthetic data, <title, body> pairs, thesauri

Cluster-based Smoothing/Scoring

- Cluster-based smoothing: (given a clustering), smooth a doc LM with a cluster of similar documents. Improves over basic LM but not significantly (Liu and Croft, 2004)
- Document expansion smoothing: smooth a doc LM with the neighboring docs (essentially one cluster per doc). Significantly improves over the basc LM (Tao et al, 2006)
- Cluster-based query likelihood. Similar to transation model, but "translate" whole doc to the query through a set of clusters (Kurland and Lee, 2004)

$$p(q|d) = \sum_{C \in clusters} p(q|C)p(C|d)$$

$p(q|C)$: likelihood of q given C
$p(C|d)$: how likely does d belong to C?

Cluster-based Smoothing/Scoring

- Cluster-based smoothing: (given a clustering), smooth a doc LM with a cluster of similar documents. Improves over basic LM but not significantly (Liu and Croft, 2004)
- Document expansion smoothing: smooth a doc LM with the neighboring docs (essentially one cluster per doc). Significantly improves over the basc LM (Tao et al, 2006)
- Cluster-based query likelihood. Similar to transation model, but "translate" whole doc to the query through a set of clusters (Kurland and Lee, 2004)

$$p(q|d) = \sum_{C \in clusters} p(q|C)p(C|d)$$

$p(q|C)$: likelihood of q given C
$p(C|d)$: how likely does d belong to C?

Cluster-based Smoothing/Scoring

- Cluster-based smoothing: (given a clustering), smooth a doc LM with a cluster of similar documents. Improves over basic LM but not significantly (Liu and Croft, 2004)
- Document expansion smoothing: smooth a doc LM with the neighboring docs (essentially one cluster per doc). Significantly improves over the basc LM (Tao et al, 2006)
- Cluster-based query likelihood. Similar to transation model, but "translate" whole doc to the query through a set of clusters (Kurland and Lee, 2004)

$$p(q|d) = \sum_{C \in clusters} p(q|C)p(C|d)$$

$p(q|C)$: likelihood of q given C
$p(C|d)$: how likely does d belong to C?

Outline

Relevance Models

Import information from docs known or assumed to be relevant

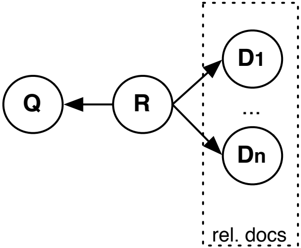
- Lavrenko, Croft 2001
- Mechanism to determine the probability $p(w|R)$ of observing a word in relevant documents
  - Query and relevant docs both sampled from an underlying relevant distribution — relevance model R underlying the information need
  - Treat docs and query as samples from R

- How to approximate a relevance model?

Relevance Models

Import information from docs known or assumed to be relevant

- Lavrenko, Croft 2001
- Mechanism to determine the probability $p(w|R)$ of observing a word in relevant documents
  - Query and relevant docs both sampled from an underlying relevant distribution — relevance model R underlying the information need
  - Treat docs and query as samples from R

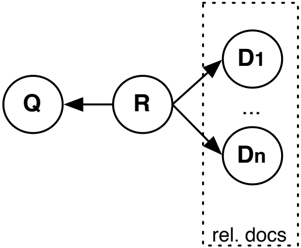


- How to approximate a relevance model?

Relevance Models

Import information from docs known or assumed to be relevant

- Lavrenko, Croft 2001
- Mechanism to determine the probability $p(w|R)$ of observing a word in relevant documents
  - Query and relevant docs both sampled from an underlying relevant distribution — relevance model R underlying the information need
  - Treat docs and query as samples from R



- How to approximate a relevance model?

Approximating Relevance Models

- Let $q = t_1, \ldots, t_k$.
  - Think of unknown process R ("black box") from which we're sampling words.
  - After sampling k times, we observe $t_1, \ldots, t_k$. What is the probability that the next word we pull out of R will be w?
  - $p(w|R) \approx p(w|t_1, \ldots, t_k)$
  - Express conditional prob. in terms of joint prob.:

$$p(w|R) \approx \frac{p(w, t_1, \ldots, t_k)}{p(t_1, \ldots, t_k)}$$

- Lavrenko and Croft use two methods to estimate $p(w, t_1, \ldots, t_k)$
  - independent sampling

$$p(w, t_1, \ldots, t_k) \propto \sum_{M \in \mathcal{M}} p(M) p(w|M) \prod_i p(t_i|M)$$

    ($\mathcal{M}$ a finite universe of unigram distributions… corresponding to docs known or assumed to be relevant)
  - conditional sampling …

Approximating Relevance Models

- Let $q = t_1, \ldots, t_k$.
  - Think of unknown process R ("black box") from which we're sampling words.
  - After sampling k times, we observe $t_1, \ldots, t_k$. What is the probability that the next word we pull out of R will be w?
  - $p(w|R) \approx p(w|t_1, \ldots, t_k)$
  - Express conditional prob. in terms of joint prob.:

$$p(w|R) \approx \frac{p(w, t_1, \ldots, t_k)}{p(t_1, \ldots, t_k)}$$

- Lavrenko and Croft use two methods to estimate $p(w, t_1, \ldots, t_k)$
  - independent sampling

$$p(w, t_1, \ldots, t_k) \propto \sum_{M \in \mathcal{M}} p(M) p(w|M) \prod_i p(t_i|M)$$

    ($\mathcal{M}$ a finite universe of unigram distributions… corresponding to docs known or assumed to be relevant)
  - conditional sampling …

Approximating Relevance Models

- Let $q = t_1, \ldots, t_k$.
  - Think of unknown process R ("black box") from which we're sampling words.
  - After sampling k times, we observe $t_1, \ldots, t_k$. What is the probability that the next word we pull out of R will be w?
  - $p(w|R) \approx p(w|t_1, \ldots, t_k)$
  - Express conditional prob. in terms of joint prob.:

$$p(w|R) \approx \frac{p(w, t_1, \ldots, t_k)}{p(t_1, \ldots, t_k)}$$

- Lavrenko and Croft use two methods to estimate $p(w, t_1, \ldots, t_k)$
  - independent sampling

$$p(w, t_1, \ldots, t_k) \propto \sum_{M \in \mathcal{M}} p(M) p(w|M) \prod_i p(t_i|M)$$

  ($\mathcal{M}$ a finite universe of unigram distributions… corresponding to docs known or assumed to be relevant)

  - conditional sampling …

Approximating Relevance Models

- Let $q = t_1, \ldots, t_k$.
    - Think of unknown process R ("black box") from which we're sampling words.
    - After sampling k times, we observe $t_1, \ldots, t_k$. What is the probability that the next word we pull out of R will be w?
    - $p(w|R) \approx p(w|t_1, \ldots, t_k)$
    - Express conditional prob. in terms of joint prob.:

$$p(w|R) \approx \frac{p(w, t_1, \ldots, t_k)}{p(t_1, \ldots, t_k)}$$

- Lavrenko and Croft use two methods to estimate $p(w, t_1, \ldots, t_k)$
    - independent sampling

$$p(w, t_1, \ldots, t_k) \propto \sum_{M \in \mathcal{M}} p(M) p(w|M) \prod_i p(t_i|M)$$

    ($\mathcal{M}$ a finite universe of unigram distributions… corresponding to docs known or assumed to be relevant)

    - conditional sampling …

Approximating Relevance Models

- Estimating $p(w, t_1, \ldots, t_k)$
  - independent sampling …
  - conditional sampling

$$p(w, t_1, \ldots, t_k) = p(w) \prod_i \sum_j p(t_i|M_j) p(M_j|w)$$

    In words: the value $p(w)$ is fixed according to some prior, then the following process is performed k times: a model M_j is selected with probability $p(M_j|w)$, then the query word t_i is sampled from M_j with probability $p(t_i|M_j)$
  - Estimate $p(M_j|w)$ using Bayes: $p(M_j|w) = p(w|M_j)p(w)/p(M_j)$
- But does it work?

Approximating Relevance Models

- Estimating $p(w, t_1, \ldots, t_k)$
  - independent sampling ...
  - conditional sampling

$$p(w, t_1, \ldots, t_k) = p(w) \prod_i \sum_j p(t_i|M_j)p(M_j|w)$$

    In words: the value $p(w)$ is fixed according to some prior, then the following process is performed k times: a model M_j is selected with probability $p(M_j|w)$, then the query word t_i is sampled from M_j with probability $p(t_i|M_j)$
  - Estimate $p(M_j|w)$ using Bayes: $p(M_j|w) = p(w|M_j)p(w)/p(M_j)$
- But does it work?

Approximating Relevance Models

- Estimating $p(w, t_1, \ldots, t_k)$
    - independent sampling ...
    - conditional sampling

$$p(w, t_1, \ldots, t_k) = p(w) \prod_i \sum_j p(t_i|M_j) p(M_j|w)$$

      In words: the value $p(w)$ is fixed according to some prior, then the following process is performed k times: a model M_j is selected with probability $p(M_j|w)$, then the query word t_i is sampled from M_j with probability $p(t_i|M_j)$
    - Estimate $p(M_j|w)$ using Bayes: $p(M_j|w) = p(w|M_j)p(w)/p(M_j)$
- But does it work?

Assessing Relevance Models

- Balog et al, SIGIR 2008, TREC Enterprise 2007 data set

- RM2 (cond samp) outperforms RM1 (ind samp) (slightly), both in a blind feedback setting (BFB-RM*) and when known relevant documents are provided (EX-RM*)

Assessing Relevance Models

- Balog et al, SIGIR 2008, TREC Enterprise 2007 data set

model	K	(possibly) relevant MAP	(possibly) relevant MRR	(highly) relevant MAP	(highly) relevant MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	10	.3145	.6326	.2679	.5335
BFB-RM2	10	.3382	.6683	.2845	.5609
EX-RM1	15	.3193	.8794	.2813	.7695
EX-RM2	25	.3454	.8596	.3111	.8169
EX-QM-ML	30	.3280	.8508	.2789	.7093
EX-QM-SM	40	.3163	.8050	.2822	.7133
EX-QM-EXP	5	.2263	.6131	.2062	.5854

Table 2: Performance of the expanded query model $\hat{Q}$.

- RM2 (cond samp) outperforms RM1 (ind samp) (slightly), both in a blind feedback setting (BFB-RM*) and when known relevant documents are provided (EX-RM*)

Assessing Relevance Models

- Balog et al, SIGIR 2008, TREC Enterprise 2007 data set

		(possibly) relevant		(highly) relevant	
model	K	MAP	MRR	MAP	MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	10	.3145	.6326	.2679	.5335
BFB-RM2	10	.3382	.6683	.2845	.5609
EX-RM1	15	.3193	.8794	.2813	.7695
EX-RM2	25	.3454	.8596	.3111	.8169
EX-QM-ML	30	.3280	.8508	.2789	.7093
EX-QM-SM	40	.3163	.8050	.2822	.7133
EX-QM-EXP	5	.2263	.6131	.2062	.5854

Table 2: Performance of the expanded query model $\hat{Q}$.

		(possibly) relevant		(highly) relevant	
model	μ	MAP	MRR	MAP	MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	0.6	.3677	.6703	.3171	.5772
BFB-RM2	0.6	.3797	.6905	.3296	.6033
EX-RM1	0.4	.4264*	.8808*	.3758*	.8259*
EX-RM2	0.4	.4273*	.9029*	.3833*	.8473*
EX-QM-ML	0.5	.4449*	.8533*	.3951*	.7911*
EX-QM-SM	0.5	.4406*	.8771*	.3955*	.8035*
EX-QM-EXP	0.7	.4016*	.8148	.3520	.7603*

Table 3: Performance of the baseline run, relevance models on blind feedback documents and sample documents, and query models on sample documents using optimal K and λ settings for each model. Results marked with * are significantly different from the baseline.

- RM2 (cond samp) outperforms RM1 (ind samp) (slightly), both in a blind feedback setting (BFB-RM*) and when known relevant documents are provided (EX-RM*)

Assessing Relevance Models

- Balog et al, SIGIR 2008, TREC Enterprise 2007 data set

		(possibly) relevant		(highly) relevant	
model	K	MAP	MRR	MAP	MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	10	.3145	.6326	.2679	.5335
BFB-RM2	10	.3382	.6683	.2845	.5609
EX-RM1	15	.3193	.8794	.2813	.7695
EX-RM2	25	.3454	.8596	.3111	.8169
EX-QM-ML	30	.3280	.8508	.2789	.7093
EX-QM-SM	40	.3163	.8050	.2822	.7133
EX-QM-EXP	5	.2263	.6131	.2062	.5854

Table 2: Performance of the expanded query model $\hat{Q}$.

		(possibly) relevant		(highly) relevant	
model	μ	MAP	MRR	MAP	MRR
baseline		.3576	.7134	.3143	.6326
BFB-RM1	0.6	.3677	.6703	.3171	.5772
BFB-RM2	0.6	.3797	.6905	.3296	.6033
EX-RM1	0.4	.4264*	.8808*	.3758*	.8259*
EX-RM2	0.4	.4273*	.9029*	.3833*	.8473*
EX-QM-ML	0.5	.4449*	.8533*	.3951*	.7911*
EX-QM-SM	0.5	.4406*	.8771*	.3955*	.8035*
EX-QM-EXP	0.7	.4016*	.8148	.3520	.7603*

Table 3: Performance of the baseline run, relevance models on blind feedback documents and sample documents, and query models on sample documents using optimal K and λ settings for each model. Results marked with * are significantly different from the baseline.

- RM2 (cond samp) outperforms RM1 (ind samp) (slightly), both in a blind feedback setting (BFB-RM*) and when known relevant documents are provided (EX-RM*)

Variation on a Theme

- Query expansion against external corpora
  - E.g., to capture different types of language usage around query
  - Diaz and Metzler, SIGIR 2006
- Start from a standard RM

$$p(t|M_q) \propto \frac{1}{|\mathcal{R}|} \sum_{d \in \mathcal{R}} p(t|M_d) p(q|M_d)$$

($\mathcal{R}$: top ranked documents)

- Take a mixture of relevance models

$$p(t|M_q) = \sum_{c \in \mathcal{C}} p(c) p(t|M_q, c)$$

($\mathcal{C}$: set of doc collections; $p(t|M_q, c)$: the relevance model computed using collection c)

- Putting it altogether

$$p(t|M_q) = \sum_{c \in \mathcal{C}} k_c \frac{p(c)}{|\mathcal{R}c|} \sum_{d \in \mathcal{R}_c} p(t|M_d) p(q|M_d)$$

Variation on a Theme

- Query expansion against external corpora
  - E.g., to capture different types of language usage around query
  - Diaz and Metzler, SIGIR 2006
- Start from a standard RM

$$p(t|M_q) \propto \frac{1}{|\mathcal{R}|} \sum_{d \in \mathcal{R}} p(t|M_d) p(q|M_d)$$

($\mathcal{R}$: top ranked documents)

- Take a mixture of relevance models

$$p(t|M_q) = \sum_{c \in \mathcal{C}} p(c) p(t|M_q, c)$$

($\mathcal{C}$: set of doc collections; $p(t|M_q, c)$: the relevance model computed using collection c)

- Putting it altogether

$$p(t|M_q) = \sum_{c \in \mathcal{C}} k_c \frac{p(c)}{|\mathcal{R}c|} \sum_{d \in \mathcal{R}_c} p(t|M_d) p(q|M_d)$$

Variation on a Theme

- Query expansion against external corpora
  - E.g., to capture different types of language usage around query
  - Diaz and Metzler, SIGIR 2006
- Start from a standard RM

$$p(t|M_q) \propto \frac{1}{|\mathcal{R}|} \sum_{d \in \mathcal{R}} p(t|M_d) p(q|M_d)$$

($\mathcal{R}$: top ranked documents)

- Take a mixture of relevance models

$$p(t|M_q) = \sum_{c \in \mathcal{C}} p(c) p(t|M_q, c)$$

($\mathcal{C}$: set of doc collections; $p(t|M_q, c)$: the relevance model computed using collection c)

- Putting it altogether

$$p(t|M_q) = \sum_{c \in \mathcal{C}} k_c \frac{p(c)}{|\mathcal{R}c|} \sum_{d \in \mathcal{R}_c} p(t|M_d) p(q|M_d)$$

Variation on a Theme

- Query expansion against external corpora
  - E.g., to capture different types of language usage around query
  - Diaz and Metzler, SIGIR 2006
- Start from a standard RM

$$p(t|M_q) \propto \frac{1}{|\mathcal{R}]} \sum_{d \in \mathcal{R}} p(t|M_d)p(q|M_d)$$

  ($\mathcal{R}$: top ranked documents)
- Take a mixture of relevance models

$$p(t|M_q) = \sum_{c \in \mathcal{C}} p(c)p(t|M_q, c)$$

  ($\mathcal{C}$: set of doc collections; $p(t|M_q, c)$: the relevance model computed using collection c)
- Putting it altogether

$$p(t|M_q) = \sum_{c \in \mathcal{C}} k_c \frac{p(c)}{|\mathcal{R}c|} \sum_{d \in \mathcal{R}_c} p(t|M_d)p(q|M_d)$$

Outline

Parsimonious Language Models

- Hiemstra et al 2004

- Stopwords at the top, typos at the bottom
- Parsimonius LMs aim to model language usage that distinguishes a relevant document from other documents

Parsimonious Language Models

- Hiemstra et al 2004

word	*probability*
the	0.0776
of	0.0386
and	0.0251
to	0.0244
in	0.0203
a	0.0198
amazon	0.0114
for	0.0109
that	0.0101
forest	0.0100
⋮	
assistence	0.0009
aleene	0.0008
macminn	0.0008
⋮	

Table 1: Example relevance model for TREC ad hoc topic 400: "amazon rain forest"

- Stopwords at the top, typos at the bottom
- Parsimonius LMs aim to model language usage that distinguishes a relevant document from other documents

Parsimonious Language Models

- Hiemstra et al 2004

word	*probability*
the	0.0776
of	0.0386
and	0.0251
to	0.0244
in	0.0203
a	0.0198
amazon	0.0114
for	0.0109
that	0.0101
forest	0.0100
⋮	
assistence	0.0009
aleene	0.0008
macminn	0.0008
⋮	

Table 1: Example relevance model for TREC ad hoc topic 400: "amazon rain forest"

- Stopwords at the top, typos at the bottom
- Parsimonius LMs aim to model language usage that distinguishes a relevant document from other documents

Parsimonious Language Models

- Three level mixture model
  - background model, relevance model, individual document model
  - $p(t_1, \ldots, t_k|d) = \prod_i((1 - \lambda - \mu)p(t_i|GE) + \mu p(t_i|R) + \lambda p(t_i|d)$
- Use the EM algorithm to train the model
  - iterative technique for estimating value of some unknown quantity, given values of correlated known quantities

1. Initialize the distribution parameters
2. Repeat until *convergence*
   1. E-step: estimate the **e**xpected value of the unkown variables, given the current parameter estimate
   2. M-step: re-estimate the distribution parameters to **m**aximize the likelihood of the data, given the expectged estimates of the unknown variables

Parsimonious Language Models

- $p(t_1, \ldots, t_k|d) = \prod_i ((1 - \lambda - \mu)p(t_i|GE) + \mu p(t_i|R) + \lambda p(t_i|d)$
  - Fixed background model, two mixture parameters
- Apply iteratively for each term t in each relevant document d, and then the M-step until estimates for $p(t|R)$ converge
  - E-step:
    $r_{t,d} = \frac{tf(t,d) \cdot \mu p(t|R)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$
    $e_{t,d} = \frac{tf(t,d) \cdot \lambda p(t|d)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$
  - M-step:
    $p(t|R) = \sum_{d \in R} \frac{r_{t,D}}{\sum_t r_{t,D}}$
    $p(t|d) = \frac{e_{t,D}}{\sum_t e_{t,D}}$
- Let's look at a slightly more understandable form and evaluate…

Parsimonious Language Models

- $p(t_1, \ldots, t_k|d) = \prod_i ((1-\lambda-\mu)p(t_i|GE) + \mu p(t_i|R) + \lambda p(t_i|d)$
  - Fixed background model, two mixture parameters
- Apply iteratively for each term t in each relevant document d, and then the M-step until estimates for $p(t|R)$ converge
  - E-step:

    $r_{t,d} = \frac{tf(t,d) \cdot \mu p(t|R)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$

    $e_{t,d} = \frac{tf(t,d) \cdot \lambda p(t|d)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$
  - M-step:

    $p(t|R) = \sum_{d \in R} \frac{r_{t,D}}{\sum_t r_{t,D}}$

    $p(t|d) = \frac{e_{t,D}}{\sum_t e_{t,D}}$
- Let's look at a slightly more understandable form and evaluate…

Parsimonious Language Models

- $p(t_1, \ldots, t_k|d) = \prod_i ((1-\lambda-\mu)p(t_i|GE) + \mu p(t_i|R) + \lambda p(t_i|d)$
  - Fixed background model, two mixture parameters
- Apply iteratively for each term t in each relevant document d, and then the M-step until estimates for $p(t|R)$ converge
  - E-step:
    $r_{t,d} = \frac{tf(t,d) \cdot \mu p(t|R)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$
    $e_{t,d} = \frac{tf(t,d) \cdot \lambda p(t|d)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$
  - M-step:
    $p(t|R) = \sum_{d \in R} \frac{r_{t,D}}{\sum_t r_{t,D}}$
    $p(t|d) = \frac{e_{t,D}}{\sum_t e_{t,D}}$
- Let's look at a slightly more understandable form and evaluate…

Parsimonious Language Models

- $p(t_1, \ldots, t_k|d) = \prod_i((1 - \lambda - \mu)p(t_i|GE) + \mu p(t_i|R) + \lambda p(t_i|d)$
  - Fixed background model, two mixture parameters
- Apply iteratively for each term t in each relevant document d, and then the M-step until estimates for $p(t|R)$ converge
  - E-step:
    $$r_{t,d} = \frac{tf(t,d) \cdot \mu p(t|R)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$$
    $$e_{t,d} = \frac{tf(t,d) \cdot \lambda p(t|d)}{(1-\lambda-\mu)p(t|GE)+\mu p(t|C)+\lambda p(t|d)}$$
  - M-step:
    $$p(t|R) = \sum_{d \in R} \frac{r_{t,D}}{\sum_t r_{t,D}}$$
    $$p(t|d) = \frac{e_{t,D}}{\sum_t e_{t,D}}$$
- Let's look at a slightly more understandable form and evaluate...

Evaluating Parsimonious Models

$$
\begin{aligned}
p(t|\hat{M}_q) &\propto p(t) \cdot \prod_{i=1}^{k} \sum_{d_j \in \mathcal{D}_q} p(q_i|d_j) \cdot p(d_j|t) \\
p(t|d_i) &= 0.5 \cdot \frac{n(t, d_j)}{\sum_{t'} n(t', d_j)} + 0.5 \cdot p(t|GE) \\
p(t|M_q) &= \lambda \cdot \frac{n(t, q)}{|q|} + (1 - \lambda) \cdot p(t|\hat{M}_q)
\end{aligned}
$$

- Make the estimate $p(t|\hat{M}_q)$ more sparse

E-step: $$e_t = n(t, d) \cdot \frac{\gamma p(t|d)}{(1 - \gamma)p(t|GE) + \gamma p(t|d)}$$

M-step: $$p(t|d) = \frac{e_t}{\sum_{t'} e_{t'}}$$

Evaluating Parsimonious Models

$$
\begin{aligned}
p(t|\hat{M}_q) &\propto p(t) \cdot \prod_{i=1}^{k} \sum_{d_j \in \mathcal{D}_q} p(q_i|d_j) \cdot p(d_j|t) \\
p(t|d_i) &= 0.5 \cdot \frac{n(t, d_j)}{\sum_{t'} n(t', d_j)} + 0.5 \cdot p(t|GE) \\
p(t|M_q) &= \lambda \cdot \frac{n(t, q)}{|q|} + (1 - \lambda) \cdot p(t|\hat{M}_q)
\end{aligned}
$$

- Make the estimate $p(t|\hat{M}_q)$ more sparse

E-step: $$e_t = n(t, d) \cdot \frac{\gamma p(t|d)}{(1-\gamma)p(t|GE) + \gamma p(t|d)}$$

M-step: $$p(t|d) = \frac{e_t}{\sum_{t'} e_{t'}}$$

Evaluating Parsimonious Models

$$p(t|\hat{M}_q) \propto p(t) \cdot \prod_{i=1}^{k} \sum_{d_j \in \mathcal{D}_q} p(q_i|d_j) \cdot p(d_j|t)$$

$$p(t|d_i) = 0.5 \cdot \frac{n(t, d_j)}{\sum_{t'} n(t', d_j)} + 0.5 \cdot p(t|GE)$$

$$p(t|M_q) = \lambda \cdot \frac{n(t, q)}{|q|} + (1 - \lambda) \cdot p(t|\hat{M}_q)$$

- Make the estimate $p(t|\hat{M}_q)$ more sparse

E-step: $$e_t = n(t, d) \cdot \frac{\gamma p(t|d)}{(1 - \gamma)p(t|GE) + \gamma p(t|d)}$$

M-step: $$p(t|d) = \frac{e_t}{\sum_{t'} e_{t'}}$$

Outline

Bringing in Explicit Knowledge

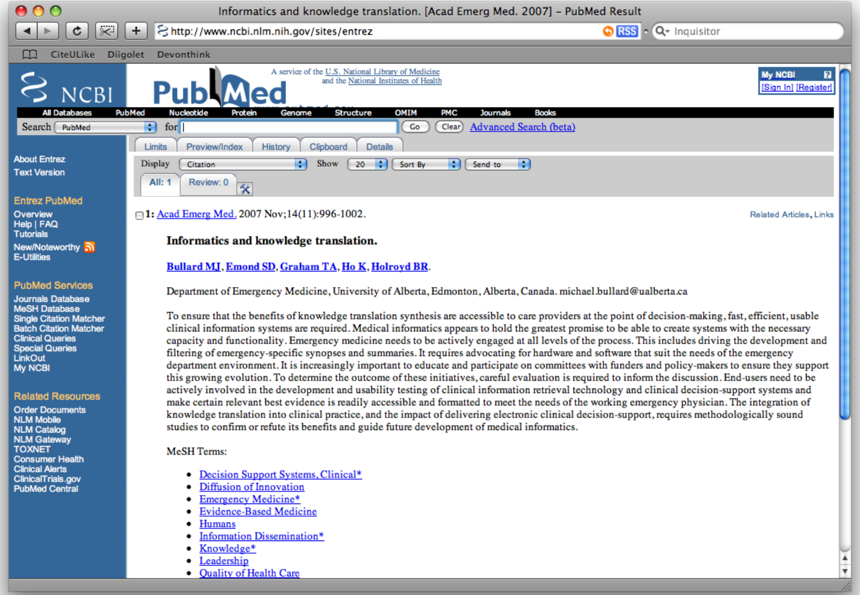
- Meij and De Rijke, 2007a, 2007b; Meij et al 2008a, 2008b
- Setting the scene
  - Digital library, e.g., scientific articles annotated with thesaurus terms
  - Access in two flavors: searching, browsing
  - How to integrate the two?
- Three step algorithm
  1. Determine the thesaurus terms most closely associated with a query
  2. Search the documents associated with these thesaurus terms, in conjunction with the query, to look for additional terms to describe the query
  3. Interpolate the query model with the found terms

Bringing in Explicit Knowledge

- Meij and De Rijke, 2007a, 2007b; Meij et al 2008a, 2008b
- Setting the scene
  - Digital library, e.g., scientific articles annotated with thesaurus terms
  - Access in two flavors: searching, browsing
  - How to integrate the two?
- Three step algorithm
  1. Determine the thesaurus terms most closely associated with a query
  2. Search the documents associated with these thesaurus terms, in conjunction with the query, to look for additional terms to describe the query
  3. Interpolate the query model with the found terms

Stop. I Need An Example

Stop. I Need An Example



Back to the Algorithm

- Three step algorithm
  1. Determine the thesaurus terms most closely associated with a query
  2. Search the documents associated with these thesaurus terms, in conjunction with the query, to look for additional terms to describe the query
  3. Interpolate the query model with the found terms

Back to the Algorithm

- For query q, rank thesaurus terms m according to

$$\begin{aligned} p(m|q) &= \frac{p(m)p(q|m)}{p(q)} \\ &\propto p(m) \sum_{d} p(q|d)p(d|m) \end{aligned}$$

- Then, estimate a thesaurus-biased relevance model by incorporating top thesaurs terms

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot p(m_1, \ldots, m_l|d)$$

- Assume m_i independent, $p(d)$ uniform:

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot \prod_{i=1,\ldots,l} p(d|m_i) \cdot P(m_i)$$

Back to the Algorithm

- For query q, rank thesaurus terms m according to

$$\begin{aligned} p(m|q) &= \frac{p(m)p(q|m)}{p(q)} \\ &\propto p(m) \sum_d p(q|d)p(d|m) \end{aligned}$$

- Then, estimate a thesaurus-biased relevance model by incorporating top thesaurs terms

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot p(m_1, \ldots, m_l|d)$$

- Assume m_i independent, $p(d)$ uniform:

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot \prod_{i=1,\ldots,l} p(d|m_i) \cdot P(m_i)$$

Back to the Algorithm

- For query q, rank thesaurus terms m according to

$$
\begin{aligned}
p(m|q) &= \frac{p(m)p(q|m)}{p(q)} \\
&\propto p(m)\sum_{d} p(q|d)p(d|m)
\end{aligned}
$$

- Then, estimate a thesaurus-biased relevance model by incorporating top thesaurs terms

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot p(m_1, \ldots, m_l|d)$$

- Assume m_i independent, $p(d)$ uniform:

$$p(w|M_q) \propto \sum_{d \in R} p(w|M_d) \cdot p(q|d) \cdot \prod_{i=1,\ldots,l} p(d|m_i) \cdot P(m_i)$$

Back to the Algorithm

- Interpolate with original query:

$$p(w|M_Q) \quad = \quad \lambda \cdot \frac{n(w,q)}{|q|} + (1-\lambda) \cdot p(w|\hat{M}_q)$$

- Evaluation
  - TREC genomics 2006
  - Passage retrieval from 160K full text biomedical docs
  - Annotated using MeSH (Medical Subject Headings)
    - 22,997 hierarchically ordered concepts; annotations by trained annotators

Retrieval Effectiveness

Comparison between different query models and a query-likelihood baseline (best scores in boldface.)

	λ	MAP		P10	
QL	1	0.359		0.45	
RM (collection)	0.10	0.426	+19%	0.48	+7%
RM (PubMed)	0.35	0.425	+18%	0.48	+7%
MM (collection)	0.05	0.424	+18%	0.48	+7%
MM (PubMed)	0.45	**0.429**	+20%	**0.49**	+9%

Retrieval Effectiveness

Comparison between different query models and a query-likelihood baseline (best scores in boldface.)

	λ	MAP		P10	
QL	1	0.359		0.45	
RM (collection)	0.10	0.426	+19%	0.48	+7%
RM (PubMed)	0.35	0.425	+18%	0.48	+7%
MM (collection)	0.05	0.424	+18%	0.48	+7%
MM (PubMed)	0.45	**0.429**	+20%	**0.49**	+9%

Evaluating the Thesaurus Terms Assigned

- Gold standard provided by TREC assessors: MeSH terms to relevant passages
- Avg agreement to assessors: 2.3/10 (RM) vs 3.0/10 (Thesaurus-biased MM); a significant difference

Evaluating the Thesaurus Terms Assigned

- Gold standard provided by TREC assessors: MeSH terms to relevant passages
- Avg agreement to assessors: 2.3/10 (RM) vs 3.0/10 (Thesaurus-biased MM); a significant difference

Relevance models		MeSH-biased models	
Collection terms	PubMed terms	Collection terms	PubMed terms
receptor	ethanol	receptor	ethanol
nicotin	nicotinic	nicotin	nicotinic
subunit	nicotine	of	nicotine
of	chronic	the	chronic
acetylcholin	cells	subunit	cells
the	treatment	**humans**	treatment
alpha7	receptor	acetylcholin	receptor
abstract	mrna	**animals**	mrna
alpha	nachr	**icotinic**	nachr
medlin	m10	**study**	m10
2003	**levels**	alpha7	subunit

Table 2. Comparison of top expansion terms for topic 173: "How do alpha7 nicotinic receptor subunits affect ethanol metabolism?", using estimations from the collection and PubMed. The terms associated with MeSH-biased models, were based on the MeSH terms as described in Table 4. Terms specific to a method are marked in boldface.

Some Further Thoughts

- Mixing the thesaurus-biased relevance model with the standard query (log)likelihood model
    - Using EM to arrive at an estimation of λ
- See work by Zhai and Lafferty, 2001, 2002

Outline

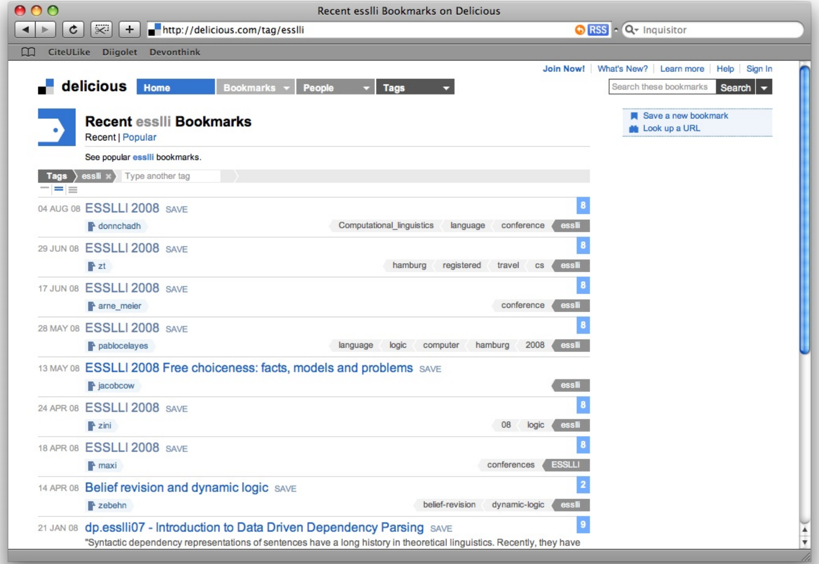
1. More Advanced Language Models
2. Relevance Models
3. Parsimonious Language Models
4. **Bringing in Explicit Knowledge**
   - Social Search
5. Importing Linguistic Features
6. Wrap Up and Look Ahead

Social Search

- Searching in a *social environment*, where a community of users actively participate in the search process
- Contrast with typical, unidirectional search engines which restrict interactions to query formulations

- Today's emphasis: tagging systems

Del.icio.us


Recent essli Bookmarks on Delicious
http://delicious.com/tag/esslli
RSS
Inquisitor
CiteULike
Diigolet
Devonthink
Join Now! | What's New? | Learn more | Help | Sign In
delicious
Home
Bookmarks
People
Tags
Search these bookmarks
Search
Recent esslli Bookmarks
Recent | Popular
See popular esslli bookmarks.
Save a new bookmark
Look up a URL
Tags
esslli
Type another tag
04 AUG 08 ESSLLI 2008 SAVE
donnchadh
Computational_linguistics language conference esslli
29 JUN 08 ESSLLI 2008 SAVE
zt
hamburg registered travel cs esslli
17 JUN 08 ESSLLI 2008 SAVE
arne_meier
conference esslli
28 MAY 08 ESSLLI 2008 SAVE
pablocelayes
language logic computer hamburg 2008 esslli
13 MAY 08 ESSLLI 2008 Free choiceness: facts, models and problems SAVE
jacobcow
esslli
24 APR 08 ESSLLI 2008 SAVE
zini
08 logic esslli
18 APR 08 ESSLLI 2008 SAVE
maxi
conferences ESSLLI
14 APR 08 Belief revision and dynamic logic SAVE
zebehn
belief-revision dynamic-logic esslli
21 JAN 08 dp.esslli07 - Introduction to Data Driven Dependency Parsing SAVE
"Syntactic dependency representations of sentences have a long history in theoretical linguistics. Recently, they have

Indexing

- Manual indexing
  - Labour intensive
  - High quality (although sometimes mistaken/limited/biased)
- Automatic indexing
  - Consistent
  - Exhaustive
- Tags
  - Low-quality in low numbers
  - Not only used for searching, but also sharing, organizing, and discovering

Tagging

- Users can add tags to anything
  - web pages
  - images
  - video's
  - etc.
- Tags provide a representation not currently provided by other sources (Heymann et al., *WSDM '08*)
- But, tags lack the size and distribution of tags necessary to make a signicant impact

Tagging

- Users can add tags to anything
    - web pages
    - images
    - video's
    - etc.
- Tags provide a representation not currently provided by other sources (Heymann et al., *WSDM '08*)
- But, tags lack the size and distribution of tags necessary to make a signicant impact

Tagging

- Users can add tags to anything
  - web pages
  - images
  - video's
  - etc.
- Tags provide a representation not currently provided by other sources (Heymann et al., *WSDM '08*)
- But, tags lack the size and distribution of tags necessary to make a signicant impact

Issues and Solutions

- Issues
  - Data sparseness
  - Vocabulary mismatch
  - Noise
- Some solutions
  - Use tags to smooth document models (Xu et al., *Advances in Knowledge Discovery and Data Mining*, 2007)
  - Map tags to WordNet categories and perform a co-occurence comparison to suggest new tags (Sigurbjörnsson and van Zwol, *WWW '08*)
  - Use tag-based features to predict tags (Heymann et al., *SIGIR '08*)

Issues and Solutions

- Issues
  - Data sparseness
  - Vocabulary mismatch
  - Noise
- Some solutions
  - Use tags to smooth document models (Xu et al., *Advances in Knowledge Discovery and Data Mining*, 2007)
  - Map tags to WordNet categories and perform a co-occurence comparison to suggest new tags (Sigurbjörnsson and van Zwol, *WWW '08*)
  - Use tag-based features to predict tags (Heymann et al., *SIGIR '08*)

Outline

A Linear Discriminant Model

- Gao et al., Linear Discriminant Model for Information Retrieval, SIGIR 2005
  - Takes into account a variety of linguistic features derived from components of a mixture model
- Start with simple unigram model, introduce hidden variables as concepts
  - $p(q|d)$ is dependent on
    - $p(c|d)$ concepts given document
    - $p(q|c,d)$ query from concept
  - Sum over all posible concepts
    - $p(q|d) = \sum_c p(q|c,d)p(c|d)$
- So what is $p(c|d)$?

A Linear Discriminant Model

- Gao et al., Linear Discriminant Model for Information Retrieval, SIGIR 2005
  - Takes into account a variety of linguistic features derived from components of a mixture model
- Start with simple unigram model, introduce hidden variables as concepts
  - $p(q|d)$ is dependent on
    - $p(c|d)$ concepts given document
    - $p(q|c,d)$ query from concept
  - Sum over all posible concepts
    - $p(q|d) = \sum_c p(q|c,d)p(c|d)$
- So what is $p(c|d)$?

A Linear Discriminant Model

- Gao et al., Linear Discriminant Model for Information Retrieval, SIGIR 2005
  - Takes into account a variety of linguistic features derived from components of a mixture model
- Start with simple unigram model, introduce hidden variables as concepts
  - $p(q|d)$ is dependent on
    - $p(c|d)$ concepts given document
    - $p(q|c,d)$ query from concept
  - Sum over all posible concepts
    - $p(q|d) = \sum_c p(q|c,d)p(c|d)$
- So what is $p(c|d)$?

Importing Linguistic Features

- Estimating $p(c|d)$
  - each concept c_i generated depending on its preceding concept c_{i-1}
  - each concept represented by its headword
  - document model is a headword bigram model: $p(h_i|h_{i-1}, d)$
- Types of concepts considered

Importing Linguistic Features

- Estimating $p(c|d)$
  - each concept c_i generated depending on its preceding concept c_{i-1}
  - each concept represented by its headword
  - document model is a headword bigram model: $p(h_i|h_{i-1}, d)$
- Types of concepts considered

Type	Models
NP	$P(\mathbf{q}' \mid \mathrm{NP}) = \mathrm{P}(h \mid \mathrm{NP}) \prod_{q \in \mathbf{q}'} P(q \mid h, \mathrm{NP})$
VP	$P(\mathbf{q}' \mid \mathrm{VP}) = \mathrm{P}(h \mid \mathrm{VP}) \prod_{q \in \mathbf{q}'} P(q \mid h, \mathrm{VP})$
NE	$P(\mathbf{q}' \mid \mathrm{NE}) = \mathrm{P}(h \mid \mathrm{NE}) \prod_{q \in \mathbf{q}'} P(q \mid h, \mathrm{NE})$, where there are three NE models, each for one type of NE.
FT	$P(\mathbf{q}' \mid \mathrm{FT}) = 1$ if $\mathbf{q}'$ can be parsed by FT grammar, 0 otherwise; where the FT grammar is a set of Finite-State Machines, each for one type of factoids

Importing Linguistic Features

- Then $p(q|d) = \sum_c p(q|c,d)p(c,d)$
  - Weights λ_i per component in this sum…
- Alternative:
  - Assume set of features $f_i(q,c,d)$ that map (q,c,d) to real values
  - Learning a function $f(q,c,d) = \{f_0(q,c,d), \ldots, f_N(q,c,d)\}$ — $N+1$ parameters
  - Relevance score:

$$Score(q,d,\vec{\lambda}) = \vec{\lambda} f(q,c,d) = \sum_{i=0,\ldots,N} \lambda_i f_i(q,c,d)$$

Importing Linguistic Features

- Then $p(q|d) = \sum_c p(q|c,d)p(c,d)$
  - Weights λ_i per component in this sum…
- Alternative:
  - Assume set of features $f_i(q,c,d)$ that map (q,c,d) to real values
  - Learning a function $f(q,c,d) = \{f_0(q,c,d), \ldots, f_N(q,c,d)\}$ — $N+1$ parameters
  - Relevance score:

$$Score(q,d,\vec{\lambda}) = \vec{\lambda} f(q,c,d) = \sum_{i=0,\ldots,N} \lambda_i f_i(q,c,d)$$

Features — Using Generative Models

- Language model scores act as feature values
  - f_0: base feature
    - $f_0(q, d) = \sum_i \mathbf{log}\, p(q_i|d)$
  - f_1 is the **log** of the bigram probability
    - $f_1(q, d) = \sum_i \mathbf{log}\, p(q_i|q_{i-1}, d)$
  - f_2 is the **log** of the doc probability
    - find chunks q' to get concepts of q, then sum probabilities
  - f_3 to f_N for remaining concepts
- New problem: how to estimate $\vec{\lambda}$
  - Paper considers two procedures
  - More on Thursday…

Features — Using Generative Models

- Language model scores act as feature values
  - f_0: base feature
    - $f_0(q, d) = \sum_i \log p(q_i|d)$
  - f_1 is the **log** of the bigram probability
    - $f_1(q, d) = \sum_i \log p(q_i|q_{i-1}, d)$
  - f_2 is the **log** of the doc probability
    - find chunks q' to get concepts of q, then sum probabilities
  - f_3 to f_N for remaining concepts
- New problem: how to estimate $\vec{\lambda}$
  - Paper considers two procedures
  - More on Thursday…

Outline

Wrap Up and Look Ahead

- Summary
  - A bit on parsimonious language models
  - Feedback and relevance models for IR
  - Modeling concepts and relations
- Tomorrow
  - On the interface of IR and IE
  - Expert finding, question answering
- On to today's practical part

Wrap Up and Look Ahead

- Summary
  - A bit on parsimonious language models
  - Feedback and relevance models for IR
  - Modeling concepts and relations
- Tomorrow
  - On the interface of IR and IE
  - Expert finding, question answering
- On to today's practical part

Wrap Up and Look Ahead

- Summary
  - A bit on parsimonious language models
  - Feedback and relevance models for IR
  - Modeling concepts and relations
- Tomorrow
  - On the interface of IR and IE
  - Expert finding, question answering
- On to today's practical part