Integrating Conceptual Knowledge into Relevance Models

A Model and an Estimation Method

Edgar Meij and Maarten de Rijke

ISLA
University of Amsterdam

1st International Conference
on the Theory of Information Retrieval (ICTIR 2007)

Universiteit van Amsterdam

Outline

Outline

1. Introduction
   - Motivation
   - Research Questions
   - Language Modeling
2. Our Algorithm
   - Overview
   - 1. Determining Concepts
   - 2. Estimating a Thesaurus-biased Model
   - 3. Interpolating the Original Query Model
3. Results and Discussion
   - Test collections
   - Retrieval Effectiveness

UNIVERSITEIT VAN AMSTERDAM

Vocabulary Gap

- A user has an information need and formulates a query
- But: not all (relevant) documents use the same term(s)
  - Different documents may contain different terms to denote a single *concept*
  - Others may even denote different *concepts* with the same term
- Solution: perform query enrichment/expansion
  - Use *conceptual* knowledge
  - Use terms from an initial result set
- **How can we combine these?**

Universiteit van Amsterdam

Vocabulary Gap

- A user has an information need and formulates a query
- But: not all (relevant) documents use the same term(s)
  - Different documents may contain different terms to denote a single *concept*
  - Others may even denote different *concepts* with the same term
- Solution: perform query enrichment/expansion
  - Use *conceptual* knowledge
  - Use terms from an initial result set
- How can we combine these?

Vocabulary Gap

- A user has an information need and formulates a query
- But: not all (relevant) documents use the same term(s)
  - Different documents may contain different terms to denote a single *concept*
  - Others may even denote different *concepts* with the same term
- Solution: perform query enrichment/expansion
  - Use *conceptual* knowledge
  - Use terms from an initial result set
- **How can we combine these?**

Research Questions

1. How can we use a language modeling framework to combine conceptual knowledge with pseudo-relevance feedback in a principled and transparent fashion?
2. Can our model compete with state-of-the-art approaches ?

Outline

Generative Language Models

- Query-likelihood approach:

$$P(d|Q) \propto P(d) \cdot \prod_{q \in Q} P(q|\theta_d)$$
$$\propto \prod_{q \in Q} \frac{c(q,d)}{|d|}$$

- With Dirichlet smoothing:

$$P(d|Q) \propto \prod_{q \in Q} \frac{c(q,d) + \mu P(q|\theta_C)}{|d| + \mu}$$

Generative Language Models

- Query-likelihood approach:

$$\begin{aligned} P(d|Q) &\propto P(d) \cdot \prod_{q \in Q} P(q|\theta_d) \\ &\propto \prod_{q \in Q} \frac{c(q,d)}{|d|} \end{aligned}$$

- With Dirichlet smoothing:

$$P(d|Q) \propto \prod_{q \in Q} \frac{c(q,d) + \mu P(q|\theta_C)}{|d| + \mu}$$

Query Models

- Generative language modeling assumes that queries are generated from documents
- Difficult to incorporate relevance feedback information
  - It's unclear how the likelihood of an "expanded query" is to be computed (as well as interpreted)
  - It's even harder to allow different query terms to have different weights
- Solution: update the query model

Query Models

- Generative language modeling assumes that queries are generated from documents
- Difficult to incorporate relevance feedback information
  - It's unclear how the likelihood of an "expanded query" is to be computed (as well as interpreted)
  - It's even harder to allow different query terms to have different weights
- Solution: update the query model

Query Models

- Ranking then comes down to calculating the *distance* between $P(w|\theta_Q)$ and $P(w|\theta_d)$ for $w \in V$
- E.g. using the KL-divergence

$$D_{kl}(\theta_Q||\theta_d) = \sum_w P(w|\theta_Q) \cdot \log \frac{P(w|\theta_Q)}{P(w|\theta_d)}$$

Query Models

- Ranking then comes down to calculating the *distance* between $P(w|\theta_Q)$ and $P(w|\theta_d)$ for $w \in V$
- E.g. using the KL-divergence

$$D_{kl}(\theta_Q||\theta_d) = \sum_w P(w|\theta_Q) \cdot \log \frac{P(w|\theta_Q)}{P(w|\theta_d)}$$

Relevance Models

- Relevance modeling assumes both the query and the document are generated from an unseen source—a relevance model
- A set of documents R is used as a model from which terms are "sampled":

$$P(w|\hat{\theta}_Q) \propto P(w) \cdot \prod_{q \in Q} \sum_{d \in R} P(q|\theta_d) \cdot P(\theta_d|w)$$

UNIVERSITEIT VAN AMSTERDAM

Relevance Models

- Relevance modeling assumes both the query and the document are generated from an unseen source—a relevance model
- A set of documents R is used as a model from which terms are "sampled":

$$P(w|\hat{\theta}_Q) \propto P(w) \cdot \prod_{q \in Q} \sum_{d \in R} P(q|\theta_d) \cdot P(\theta_d|w)$$

Outline

Our Algorithm

Our algorithm assumes every document is annotated with one or more *concepts*

- tags
- classifications/classes
- thesaurus terms

Our Algorithm: Three Steps

1. Determine the concepts most closely associated with a query
2. Look at the documents associated with these concepts, in conjunction with the query, to establish a *conceptually-biased* (or thesaurus-biased) relevance model
3. Interpolate the original query model with the found terms

Determining Concepts

For a given query Q, rank the concepts $m \in M$ according to:

$$\begin{aligned} P(m|Q) &= \frac{P(m) \cdot P(Q|m)}{P(Q)} \\ &= P(m) \cdot \sum_{d} P(Q|d) \cdot P(d|m) \end{aligned}$$

Universiteit van Amsterdam

Estimating a Thesaurus-biased Model

- Then, estimate a thesaurus-biased relevance model by incorporating the top-l thesaurus terms
- Assuming the thesaurus terms $m_1, \ldots, m_l$ to be independent, we obtain

$$\begin{aligned}
& P(w|\hat{\theta}_Q) \\
& \quad \propto \quad P(w) \cdot \prod_{q \in Q} \sum_{d \in R} P(q|\theta_d) \cdot P(w|\theta_d) \cdot P(m_1, \ldots, m_l|\theta_d) \\
& \quad \propto \quad P(w) \cdot \prod_{q \in Q} \sum_{d \in R} P(q|\theta_d) \cdot P(w|\theta_d) \cdot \prod_{i=1}^{l} P(d|m_i) \cdot P(m_i)
\end{aligned}$$

Interpolating the Original Query Model

- Finally, the found model is interpolated with the original query (using a mixing weight λ) to yield the final query model

$$P(w|\theta_Q) \quad = \quad \lambda \cdot P(w|\tilde{\theta}_Q) + (1-\lambda) \cdot P(w|\hat{\theta}_Q),$$

where

$$P(w|\tilde{\theta}_Q) \quad = \quad \frac{c(w,Q)}{|Q|}.$$

- When λ is set to 1, a query-likelihood ranking is obtained

Interpolating the Original Query Model

- Finally, the found model is interpolated with the original query (using a mixing weight λ) to yield the final query model

$$P(w|\theta_Q) = \lambda \cdot P(w|\tilde{\theta}_Q) + (1 - \lambda) \cdot P(w|\hat{\theta}_Q),$$

where

$$P(w|\tilde{\theta}_Q) = \frac{c(w, Q)}{|Q|}.$$

- When λ is set to 1, a query-likelihood ranking is obtained

Estimating λ

- To find λ, we approximate the query model space using (pseudo-)relevant documents
- The log of the likelihood of observing these terms is:

$$\log P(w_1, \ldots, w_n|\theta_Q, \lambda) = \sum_i \pi_i \prod_{j=1}^{n} \log \left(\lambda P(w_j|\tilde{\theta}_{q_i}) + (1-\lambda)P(w_j|\hat{\theta}_{q_i})\right)$$

- Then, use the EM algorithm to find λ which maximizes this loglikelihood:

$$\lambda^* = \arg\max_{\lambda} \log P(w_1, \ldots, w_n|\theta_Q).$$

Estimating λ

- To find λ, we approximate the query model space using (pseudo-)relevant documents
- The log of the likelihood of observing these terms is:

$$\log P(w_1, \ldots, w_n | \theta_Q, \lambda) = \sum_i \pi_i \prod_{j=1}^{n} \log \left(\lambda P(w_j | \tilde{\theta}_{q_i}) + (1 - \lambda) P(w_j | \hat{\theta}_{q_i}) \right)$$

- Then, use the EM algorithm to find λ which maximizes this loglikelihood:

$$\lambda^* = \arg\max_{\lambda} \log P(w_1, \ldots, w_n | \theta_Q).$$

Estimating λ

Update formulas:

$$\pi_i^{k+1} = \frac{\pi_i^k \prod_{j=1}^n \left(\lambda^k P(w_j|\tilde{\theta}_{q_i}) + (1-\lambda^k) P(w_j|\hat{\theta}_{q_i}\right)}{\sum_{i'} \pi_{i'}^k \prod_{j=1}^n (\lambda^k P(w_j|\tilde{\theta}_{q_{i'}}) + (1-\lambda^k) P(w_j|\hat{\theta}_{q_{i'}})}$$

$$\lambda^{k+1} = \frac{1}{n} \sum_i \pi_i^{k+1} \sum_{j=1}^n \frac{\lambda^k P(w_j|\tilde{\theta}_{q_i})}{\lambda^k P(w_j|\tilde{\theta}_{q_i}) + (1-\lambda^k) P(w_j|\hat{\theta}_{q_i})}$$

Outline

TREC Genomics

	Size	Vocab. size
trecgen05	4,591,008 **abstracts**	800,477,879
trecgen06	162,259 **full-text docs**	1,090,232,994

- PubMed
  - Bibliographic database maintained by the National Library of Medicine (NLM)
  - Over 15M entries, containing author information, abstracts, etc. etc.
- Medical Subject Headings (MeSH)
  - 22,997 hierarchically ordered concepts
  - Trained annotators from the NLM assign one or more MeSH terms to every document indexed in PubMed

TREC Genomics

	Size	Vocab. size
trecgen05	4,591,008 **abstracts**	800,477,879
trecgen06	162,259 **full-text docs**	1,090,232,994

- PubMed
  - Bibliographic database maintained by the National Library of Medicine (NLM)
  - Over 15M entries, containing author information, abstracts, etc. etc.
- Medical Subject Headings (MeSH)
  - 22,997 hierarchically ordered concepts
  - Trained annotators from the NLM assign one or more MeSH terms to every document indexed in PubMed

TREC Genomics

	Size	Vocab. size
trecgen05	4,591,008 **abstracts**	800,477,879
trecgen06	162,259 **full-text docs**	1,090,232,994

- PubMed
  - Bibliographic database maintained by the National Library of Medicine (NLM)
  - Over 15M entries, containing author information, abstracts, etc. etc.
- Medical Subject Headings (MeSH)
  - 22,997 hierarchically ordered concepts
  - Trained annotators from the NLM assign one or more MeSH terms to every document indexed in PubMed

Retrieval Effectiveness

MAP	QL	RM		MM	
trecgen05	0.218	0.220	+2.33%	**0.241**	+12.09%
trecgen06	0.359	0.360	+0.28%	**0.416**	+15.88%

P10	QL	RM		MM	
trecgen05	0.369	**0.374**	+1.36%	0.360	-2.44%
trecgen06	0.450	0.454	+0.89%	**0.465**	+3.33%

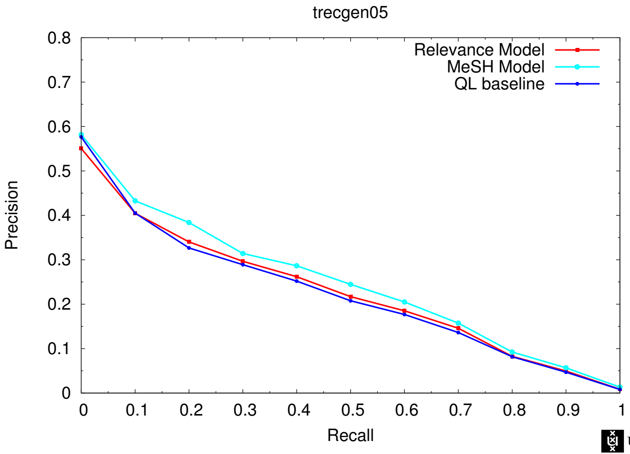
Ad-hoc retrieval results of the query-likelihood baseline (QL), Relevance model (RM), and thesaurus-biased model (MM) (best scores in boldface.)

Retrieval Effectiveness

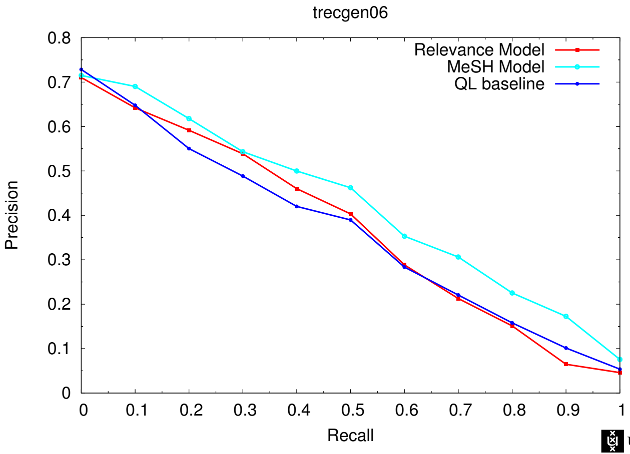
MAP	QL	RM		MM	
trecgen05	0.218	0.220	+2.33%	**0.241**	+12.09%
trecgen06	0.359	0.360	+0.28%	**0.416**	+15.88%
P10	QL	RM		MM	
trecgen05	0.369	**0.374**	+1.36%	0.360	-2.44%
trecgen06	0.450	0.454	+0.89%	**0.465**	+3.33%

Ad-hoc retrieval results of the query-likelihood baseline (QL), Relevance model (RM), and thesaurus-biased model (MM) (best scores in boldface.)

Retrieval Effectiveness



Retrieval Effectiveness



Summary and Future Work

Summary

- We have shown how to incorporate conceptual knowledge into a language modeling framework, through biasing relevance models
- We have used the EM algorithm to estimate λ and, using this approach, our model outperforms both a query-likelihood baseline, as well as state-of-the-art relevance models on two distinct test collections

Future Work

- Incorporate structure/relations between concepts
- Move to different domain(s) with different annotations