Authors: Huang, Croft
Presenter: Teodor Stoev

A Unified Relevance Model for Opinion Retrieval

Opinion Retrieval

- What is Opinion Retrieval (OR)?
  - Rank a set of documents with respect to
    - topic relevance
    - sentiment
  - Retrieve and analyze online opinions precisely and efficiently
- Why do we need OR?
  - People share their personal opinions and reviews about consumer products, commercial services, politics, etc.
  - Their opinions
    - help other people to make decisions
    - help business and government agencies to collect valuable feedback

What is the problem with OR

- Different from traditional topic-based retrieval
  - Relevant documents should at the same time
    - be relevant to the topic
    - contain subjective opinions
  - Text collections are more informal web data
    - E.g. blogs, forums

What is the problem with OR

- Modeling the ***information need*** is the greatest challenge
  - Defined by a user query consisting of a small number of keywords
    - content words only
      - E.g. “Steve Jobs”
    - content words + cue words
      - E.g. “Find opinions about iPhone”
  - Using these queries directly leads to very low precision
    - They need to be enhanced!

Approaches to OR

- Two-stage ranking process
  - Documents are ranked by topical relevance only
  - Candidate relevant documents are then re-ranked by their opinion scores
  - High computational overhead
- Unified ranking process
  - Unify topic retrieval and sentiment classification into a single process
  - Potentially better performance
- What is the paper about?
  - A **unified ranking** approach
  - An efficient OR procedure based on query expansion

BACKGROUND

Approaches to Information Retrieval

- The Basic Question
  - "What is the probability that *this* document is relevant to *this* query"
  - Random variables D (document), Q (query), R (relevance)

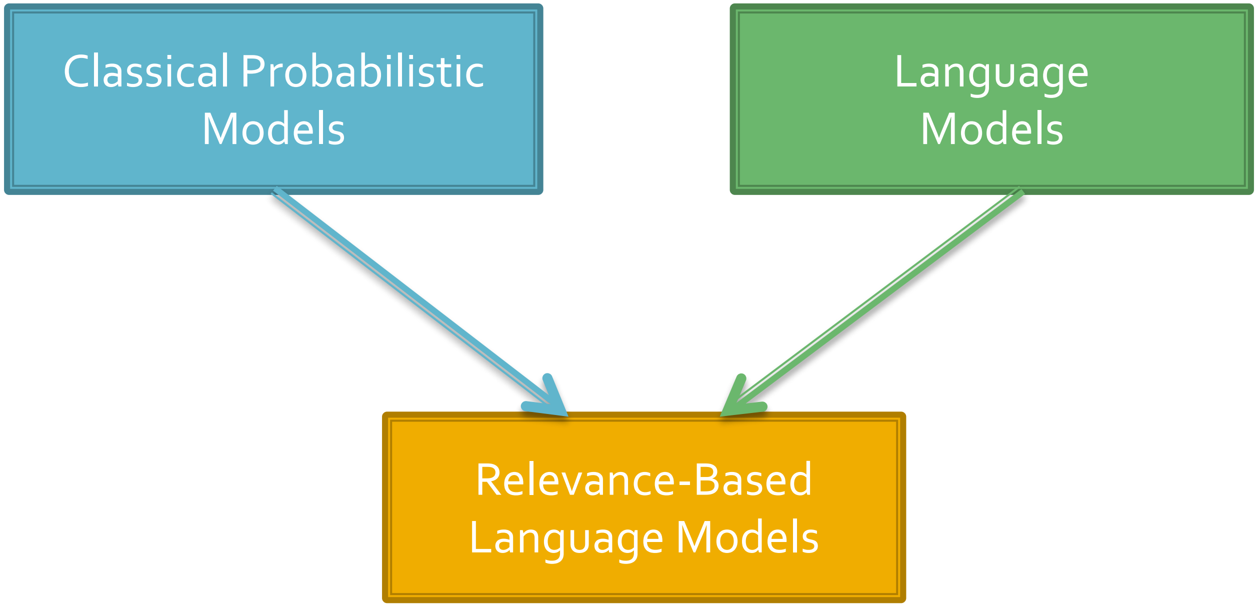
$$p(R = r \mid D, Q) \quad = \quad 1 - p(R = \bar{r} \mid D, Q)$$

- A *generative* relevance model

$$p(R = r \mid D, Q) \quad = \quad \frac{p(D, Q \mid R = r)\, p(R = r)}{p(D, Q)}$$

- Two different ways to evaluate relevance probability
  - Use explicit models of relevance (Classical Probabilistic Models)
  - Model query generation (Language Models)

Approaches to Information Retrieval



Classical Probabilistic Models

- The ***probability ranking principle***
  - rank the documents *D* by the odds of their being observed in the relevant class:

$$P(D|R)/P(D|N)$$

  - R – class of documents relevant to a query
  - N – class of non-relevant documents

Classical Probabilistic Models

- Ranking the documents
  - The Binary Independence Model

$$P(D|R) = \prod_{w \in D} P(w|R) \prod_{w \notin D} (1 - P(w|N))$$

    - document = binary vector over the entire vocabulary
    - $P(w|R)$ – the probability of a word ω being present in a document sampled from the relevant class
  - A common word independence assumption

$$\frac{P(D|R)}{P(D|N)} \sim \prod_{w \in D} \frac{P(w|R)}{P(w|N)}$$

    - document = sequence of words

Classical Probabilistic Models

- What is a ***relevance model***?
  - a mechanism to determine the probability $P(w|R)$ of observing a word ω in the documents relevant to a particular information need
- Estimating the relevance model
  - a primary obstacle when no training data is available

Language Models

- What is a ***language model***?
  - a probability distribution that assigns a probability to a sequence of words
- What do we use language models for?
  - Speech recognition:
    - a language model captures the properties of a language
    - tries to predict the next word in a speech sequence
  - Information retrieval:
    - a language model is generated for every document in a collection
    - documents are ranked based on the probability that the document's language model would generate the terms of the query

Language Models

- Ponte and Croft's Model
  - Query *Q* – a binary vector over the entire vocabulary

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D) \prod_{w \notin Q} (1 - P(w|M_D))$$

  - The major difference: estimation of individual word probabilities
  - A *smoothed* non-parametric estimate for $P(w|M_D)$
- A sequence of independent words instead of a binary vector

$$P(Q|M_D) = \prod_{w} P(w|M_D)^{q_w}$$

  - q_w – the number of times the word *ω* occurs in the query

Relevance-Based Language Models

- Estimating the relevance model
  - with training data (relevance judgments)
    - count the number of occurrences of ω in the relevant documents and appropriately *smooth* the counts
  - without training data
    - using the query alone
- The idea
  - Incorporate pseudo-relevance feedback into a language modeling approach

Approximating a Relevance Model

- Construct the probability distribution $P(w|R)$
  - Relate the probability of *ω* to the conditional probability of observing *ω* given that we just observed $q_1 \ldots q_k$

$$P(w|R) \approx P(w|q_1 \ldots q_k)$$

$$P(w|R) \approx \frac{P(w, q_1 \ldots q_k)}{P(q_1 \ldots q_k)}$$

  - The challenge: estimate the joint probability $P(w, q_1 \ldots q_k)$
    - *i.i.d.* sampling
    - Conditional sampling

Smoothing

- What is ***smoothing***?
  - To smooth a data set is to create an approximating function that attempts to capture important patterns in the data, while leaving out noise
  - Usually used in statistics and image processing
  - When using *Dirichlet* distribution as the prior for smoothing, we get *additive smoothing*

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d)$$

    - $\alpha > 0$: smoothing parameter

Back to our topic

OPINION RETRIEVAL

Lexicon-based opinion finding

- Sentiment lexicon-based methods lead to good performance in two-stage opinion retrieval
- A lightweight lexicon-based statistical approach
  - The distribution of terms in relevant opinionated documents is compared to their distribution in relevant fact-based documents to calculate an opinion weight
  - The weights are then used to compute opinion scores for every retrieved document
  - A weighted dictionary is generated from previous TREC relevance data
  - The dictionary is then submitted as a query to a search engine to get an initial query-independent opinion score of all retrieved documents

Unified sentiment retrieval model

- A key issue in Opinion Retrieval
  - how to combine a document's opinionate score and topic relevance score
- Previous research
  - tries to unify topic and opinion relevance by modeling document generation
- Problems encountered so far
  - Intensive computation load inevitable during retrieval
    - for each possible candidate document, an opinion score summed up from the generative probability of thousands of sentiment words

Opinion Relevance Model

- Extends the ***relevance model*** of Lavrenko and Croft
- We assume that we can obtain an ***opinion relevance model*** from a query
- Notation
  - R : the opinion relevance model for a query
  - D : the document model
  - V : the vocabulary
- V is divided into two separate vocabularies
  - CV : Content word vocabulary
  - OV : Opinion word vocabulary

Opinion Relevance Model

- Compare the query to the documents
  - ***Kullback-Leibler (KL)-divergence*** between the probability distributions of the opinion relevance model and the document model

$$KL(R||D) = \sum_{w \in V} P(w|R) log \frac{P(w|R)}{P(w|D)}$$
$$= \sum_{w \in V} P(w|R) log P(w|R) - \sum_{w \in V} P(w|R) log P(w|D)$$

  - $\sum_{w \in V} P(w|R) log P(w|R)$ is identical for all documents
- Document scores

$$Score(D) = \sum_{w \in V} P(w|R) log P(w|D)$$

Opinion Relevance Model

- *Dirichlet* smoothing

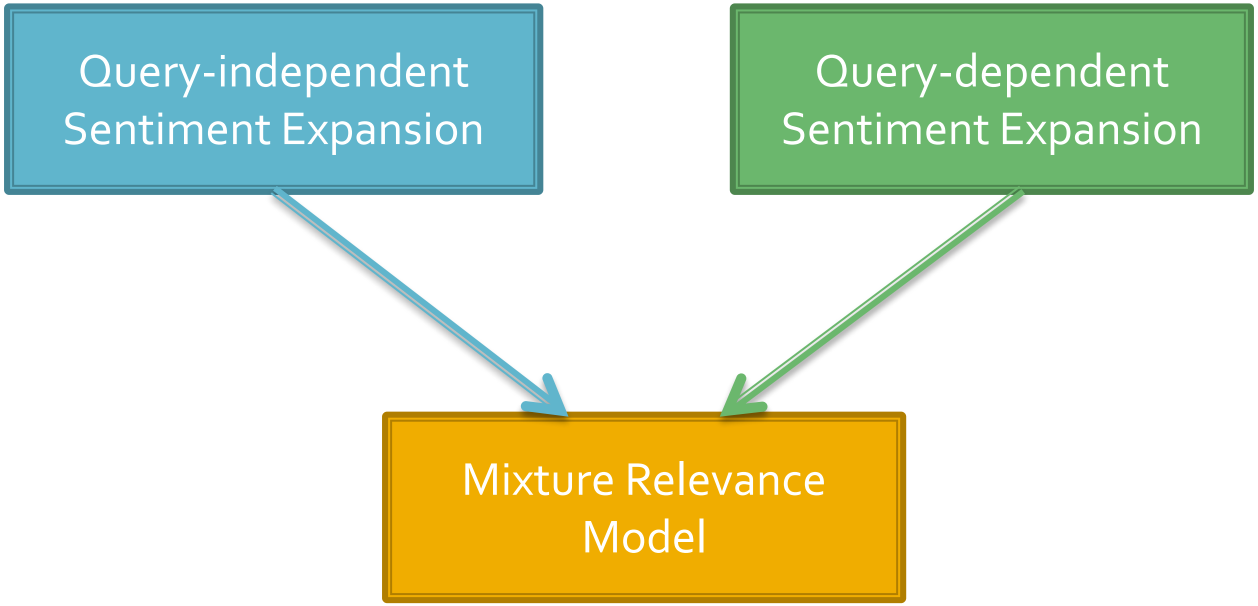
$$P(w|D) = \frac{freq_D(w) + \mu C_w/|C|}{|D| + \mu}$$

- A unified model of both topic relevance and sentiment

$$Score(D) = \alpha \sum_{w \in CV} P(w|R) log P(w|D) +$$
$$+ (1 - \alpha) \sum_{w \in OV} P(w|R) log P(w|D)$$

  - α balances the two relevant scores

Sentiment Expansion Techniques



Query-independent Sentiment Expansion

- Based on seed words
  - restricts *opinion vocabulary* to some predefined seed words
    - Positive seeds: good, nice, excellent, positive, fortunate, correct, superior
    - Negative seeds: bad, nasty, poor, negative, unfortunate, wrong, inferior
- Based on text corpora
  - obtain opinion words from lexical resources
    - thousands of entries in a lexicon
    - only the most frequent opinion words selected to expand the original queries

Query-independent Sentiment Expansion

- Based on relevance data
  - Learn a function automatically from training data to rank documents
  - Given a set of query relevance judgments, we define a ***contribution*** of an opinion word *ω*
    - the maximum increase in the mean average precision (MAP) of the expanded queries over a set of original queries
    - *ω is* used to expand every original query
  - Contribution of *ω* = how much *ω* can improve the performance of opinion retrieval
  - The words with the highest contribution are used for sentiment expansion

Query-dependent Sentiment Expansion

- A target is often associated with some particular opinion words
  - e.g. Mozart -> genius , famous
- The idea
  - Extract opinion words from a set of user-provided relevant opinionated documents
- What if there are no user-provided relevant documents?
  - The relevant document set *C* can be acquired through pseudo-relevance feedback
    - rank documents using query likelihood scores
    - select some top ranked documents to get the pseudo relevant set of *C*

Query-dependent Sentiment Expansion

- Condition the probability of ω on a query

$$P(w|R) \approx P(w|Q) = P(w|q_1, q_2, \ldots, q_n)$$

- A relevance feedback method to extract opinion words from a set of user-provided relevant opinionated documents

$$P(w|q_1, q_2, \ldots, q_n) = \frac{P(w, q_1, q_2, \ldots, q_n)}{P(q_1, q_2, \ldots, q_n)}$$

$$P(q_1, q_2, \ldots, q_n) = \sum_{w} P(w, q_1, q_2, \ldots, q_n)$$

Query-dependent Sentiment Expansion

- The joint probability estimated given the relevant document set of C

$$P(w, q_1, q_2, \ldots, q_n) = \sum_{D \in C} P(D) P(w, q_1, q_2, \ldots, q_n | D)$$

$$P(w, q_1, q_2, \ldots, q_n | D) = P(w|D) P(q_1, q_2, \ldots, q_n | D, w)$$

- Assuming q_i is conditionally independent from q_j

$$P(q_1, q_2, \ldots, q_n | D, w) \approx \prod_{i=1}^{n} P(q_i | D, w)$$

$$P(q_i|D, w) = \begin{cases} freq_D(q_i)/|D| & \text{if } w \text{ occurs in } D \\ 0 & \text{otherwise} \end{cases}$$

Mixture Relevance Model

- Final score of a document

$$Score(D) = \alpha \sum_{w \in Q} P(w|Q) log P(w|D) + \\ + \beta \sum_{w \in OV_1} P(w|R_1) log P(w|D) + \\ + (1 - \alpha - \beta) \sum_{w \in OV_2} P(w|R_2) log P(w|D)$$

  - Interpolation of the scores assigned by
    - original query
    - query-independent sentiment expansion
    - query-dependent sentiment expansion

Experimental setup

- Benchmark collections

Collection		Blog06	COAE08
Evaluation		TREC/Blog	COAE
Topic	Training	50	Not available
	Testing	100	20
	Example	*Mozart*	李连杰(*Jet Li*)
Documents	Number	3215K	40K
	Size	20GB	52M

- Sentiment resources
  - English
    - External sentiment lexicons – several thousand opinion words
  - Chinese
    - A sentiment lexicon of about 7,000 opinion words

Experimental setup

- Sentiment resources

Corpus	Opinionated collection			General collection	
	Movie	MPQA	Blog06(op)	Blog06	Web
Documents	2000	535	11523	3.2M	14B
Most frequent English opinion words	like	against	like	like	free
	even	minister	know	complete	back
	good	terrorism	even	good	like
	too	even	good	know	best
	plot	like	too	free	show

Word	Average TF	DF	Coverage
like	8.32	8200	71.2%
know	5.27	6970	60.5%
even	4.67	6521	56.6%
good	4.59	7047	61.2%
too	3.12	5998	52.1%

Results – sentiment expansion

- Comparison of different sentiment expansion approaches

Approach			Evaluation metric			
Category	Sub-category	Run id	MAP	R-prec	bPref	P@10
Baseline	/	Baseline	0.2655	0.3252	0.2974	0.4770
Query-independent	Seed words	Seed-1	0.2797	0.3335	0.3120	0.5250
		Seed-7	0.2650	0.325	0.3058	0.4690
	Opinionated corpus	Movie	0.2961	0.3422	0.3303	0.5460
		MPQA	0.2732	0.3315	0.3082	0.4880
		Blog06(op)	0.3097	0.3530	0.3395	0.5570
	General corpus	Blog06	0.2822	0.3340	0.3133	0.5200
		Web	0.2733	0.3313	0.3055	0.5100
	Relevance data	RD	0.3117	0.3542	0.3408	0.5650
Query-dependent	Pseudo relevance feedback	PRB	0.2806	0.3333	0.3101	0.4950
Mixture model	Blog06(op)+PRB	MBoP	0.3124	0.3521	0.3404	**0.5670**
	Blog06+PRB	MBP	0.3009	0.3477	0.3340	0.5480
	RD+PRB	MRDP	**0.3147**	**0.3546**	**0.3418**	0.5640

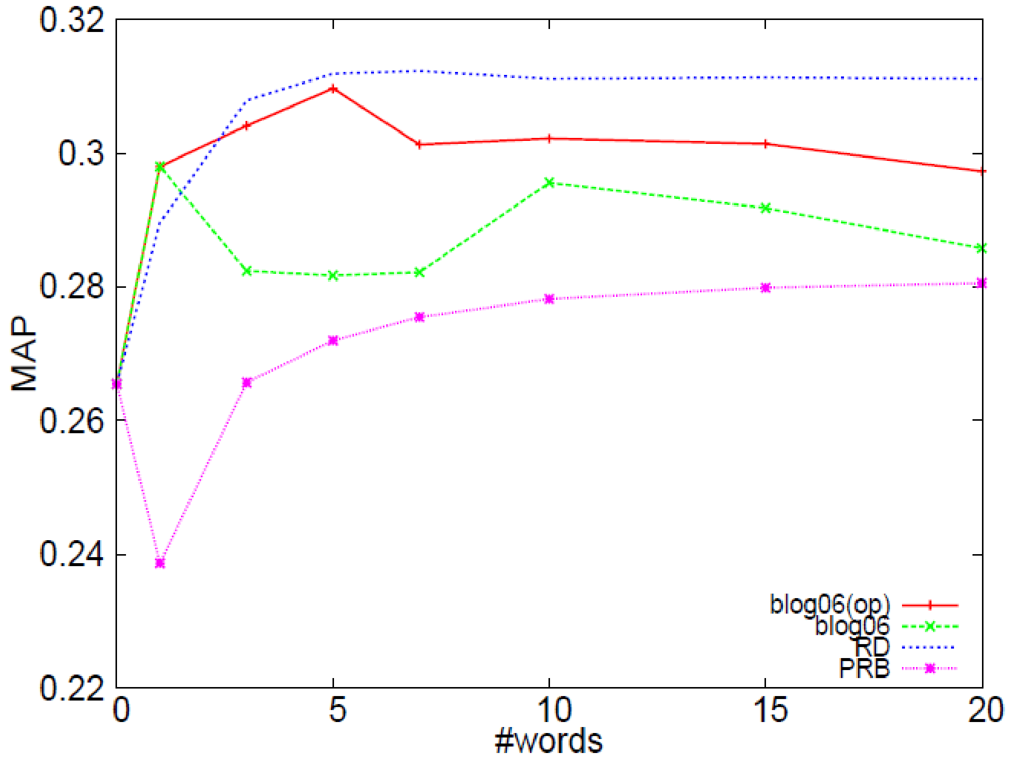
Results – sentiment expansion

- Comparison of different sentiment expansion approaches

Approach			Evaluation metric			
Category	Sub-category	Run id	MAP	R-prec	bPref	P@10
Baseline	/	Baseline	0.3565	0.4046	0.3874	0.7700
Query-independent	Opinionated corpus	Product	0.3597	0.4149	0.3932	0.7750
		Hotel	0.3658	0.4240	0.4011	0.7700
	General corpus	COAE08	0.3621	0.4174	0.3959	0.7550
		Sougo	0.3571	0.4139	0.3880	0.7700
Query-dependent	Pseudo relevance feedback	PRB	0.3677	0.4273	0.4031	0.7600
Mixture model	Hotel+PRB	MHP	**0.3697**	**0.4311**	**0.4069**	0.7750
	COAE08+PRB	MCP	0.3685	0.4286	0.4060	**0.7900**

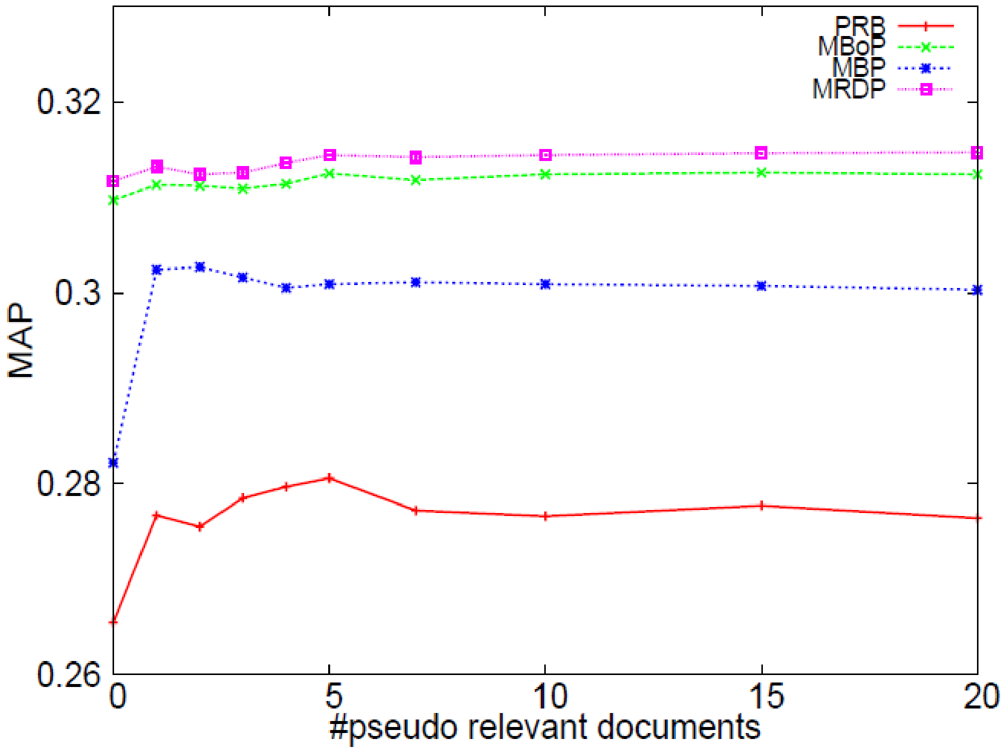
Results – number of opinion words

- Results of varying the number of opinion words



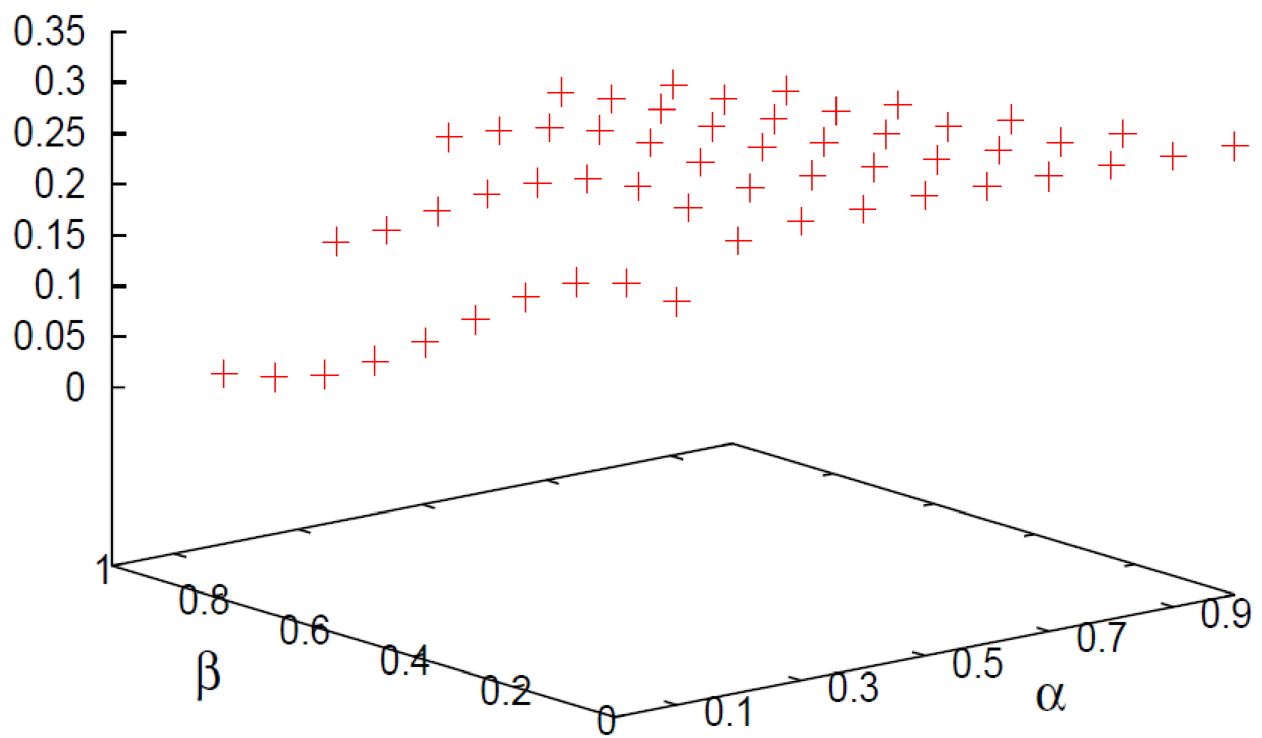
Results – number of pseudo-relevant documents

- Results of varying the number of pseudo-relevant documents



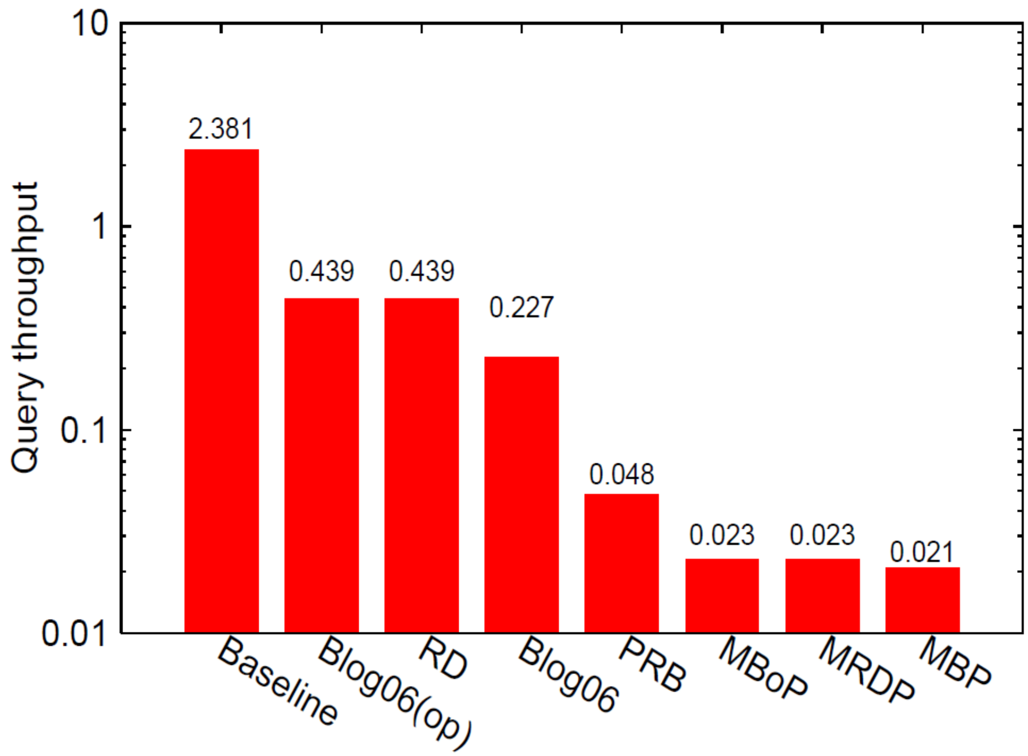
Results – Parameter estimation

- MAP surface over simplex of parameter values



Results – Query throughput

- Query throughput of different sentiment expansion approaches



Conclusion

- *Opinion relevance model* – a formal framework for directly modeling the information need for opinion retrieval
- Studies do far (two-stage process)
  - A strongly performing topic-based retrieval baseline is very important in achieving good opinion retrieval performance
- New results
  - Good opinion retrieval also improves topic-based retrieval
- Future work
  - Consider the diversity of the feedback documents
  - Vary the document priors
  - Take into account the mechanics of blogs and forums

Open Issues

- Whether a word is opinionated or not is still debatable
    - E.g. “home”, “just”, “so”
- Semantic orientation depends on context
    - Positive and negative orientations of “long”
        - “The battery of this camera lasts very long”
        - “This program takes a long time to run”
- Explicit and implicit opinions
    - **Explicit** positive opinion
        - “The picture quality of this camera is amazing.”
    - **Implicit** negative opinions
        - “The earphone broke in two days.”

References

- Huang, Croft. A Unified Relevance Model for Opinion Retrieval. 2010.
- Lafferty, Zhai. Probabilistic Relevance Models Based on Document and Query Generation. 2002.
- Ponte, Croft. A Language Modeling Approach to Information Retrieval. 1998.
- Lavrenko, Croft. Relevance Based Language Models. 2001.
- Ding, Liu, Yu. A Holistic Lexicon-Based Approach to Opinion Mining. 2008.