A Pólya Urn Document Language Model for Information Retrieval

Ronan Cummins

Cambridge, 2014

Outline

Quote

"An author writes not only by processes of *association* – i.e. sampling earlier segments of the word sequence – but also by process of *imitation* – i.e. sampling segments of word sequences from other works he has written, from works of other authors, and, of course, from sequences he has heard."

— Herbert A. Simon (1955)

Motivation

- Bag of words
- Term-dependencies
  - Improves retrieval effectiveness +
  - Leads to more complex models -
  - ClueWeb09 (1 Billion documents)

Motivation

- Bag of words
- Term-dependencies
  - Improves retrieval effectiveness +
  - Leads to more complex models -
  - ClueWeb09 (1 Billion documents)
- Can we create a retrieval model that includes dependencies but without any additional cost?

Two Kinds of Term Dependency

Examples

Traditional dependencies

- *Captain Beefheart*
- *Che Guevara*

Two Kinds of Term Dependency

Examples

Traditional dependencies

- *Captain Beefheart*
- *Che Guevara*

Examples

Word Burstiness

- A different kind of dependency
- *"Cycling on the footpath is dangerous. A footpath is ..."*
- Synonyms: {footpath, pavement, sidewalk }
- Preference for the word already used

Word Burstiness

- Initial choice of a word to describe a *'concept'* affects subsequent usage
- The tendency of an otherwise rare word to occur multiple times in a document (Church, 1995; Madsen; 2005)
- A form of *preferential attachment* (e.g. 'the rich get richer')
- A generative language model that includes preferential attachment better explains Zipfian (power-law) characteristics (Simon, 1955; Mitzenmacher, 2004)
- Two-stage language models (Goldwater et al, 2011)

Outline

VSM

	cat	dog	footpath	animal	hot	mat	Documents
	0	5	0	5	0	7	d1
	1	2	4	0	2	0	d2
	0	1	0	0	1	0	Query

Figure : vector space example

Tradition

- Place documents and queries in a multidimensional term space
- Use measures of closeness in the space as measures of similarity
- Conceptually useful
- But?

Tradition

- Place documents and queries in a multidimensional term space
- Use measures of closeness in the space as measures of similarity
- Conceptually useful
- But?
- What weights to use?
- What matching function to use?
- Experiments tell us that cosine matching function is very poor
- linear tf and idf has very poor performance
- What did we gain from the VSM other than an inner-product matching function?

Outline

Language Modelling for Retrieval

- First approaches appeared in 1998 (Ponte and Croft, 1998; Hiemstra, 1998)
- Relevance-based approaches (Lavrenko, 2001)
- Studies of smoothing (Zhai and Lafferty, 2001)
- Dirichlet compound multinomial relevance language model (Xu and Akella, 2008)
- Positional language models (Lv and Zhai, 2009)
- State-of-the-art unigram model uses a Dirichlet prior on the background multinomial updated with a document (Zhai and Lafferty, 2004)

Language Modelling for Retrieval

- First approaches appeared in 1998 (Ponte and Croft, 1998; Hiemstra, 1998)
- Relevance-based approaches (Lavrenko, 2001)
- Studies of smoothing (Zhai and Lafferty, 2001)
- Dirichlet compound multinomial relevance language model (Xu and Akella, 2008)
- Positional language models (Lv and Zhai, 2009)
- State-of-the-art unigram model uses a Dirichlet prior on the background multinomial updated with a document (Zhai and Lafferty, 2004)

Query-Likelihood Model

- Rank documents d in order of the likelihood of their model $\mathcal{M}_d$ generating the query string q
- General ranking principle for a probabilistic language model

Query-Likelihood Model

- Rank documents d in order of the likelihood of their model $\mathcal{M}_d$ generating the query string q
- General ranking principle for a probabilistic language model

$$p(q|\mathcal{M}_d = \theta_{dm}) = \prod_{t \in q} p(t|\theta_{dm})^{c(t,q)} \tag{1}$$

Query-Likelihood Model

- Rank documents d in order of the likelihood of their model $\mathcal{M}_d$ generating the query string q
- General ranking principle for a probabilistic language model

$$p(q|\mathcal{M}_d = \theta_{dm}) = \prod_{t \in q} p(t|\theta_{dm})^{c(t,q)} \tag{1}$$

$$log\ p(q|\mathcal{M}_d = \theta_{dm}) = \sum_{t \in q} (log\ p(t|\theta_{dm}) \cdot c(t,q)) \tag{2}$$

Query Likelihood

Documents

	cat	dog	footpath	animal	hot	mat
d1	0	5	0	5	0	7
d2	1	2	4	0	2	0
Query	0	1	0	0	1	0

Query Likelihood

	cat	dog	footpath	animal	hot	mat	Documents
	0	5/17	0	5/17	0	7/17	d1
	1/9	2/9	4/9	0	2/9	0	d2

Query = {hot dog}

Query Likelihood

Documents

	cat	dog	footpath	animal	hot	mat
d1	0	5/17	0	5/17	0	7/17
d2	1/9	2/9	4/9	0	2/9	0

Query = {hot dog}

Zero probabilities are especially problematic for longer queries

Smoothing I

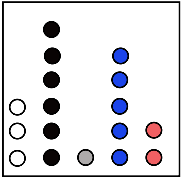
- Avoids over-fitting

$$p(t|\hat{\theta}_{dm}) = (1 - \pi) \cdot p(t|\hat{\theta}_d) + \pi \cdot p(t|\hat{\theta}_c) \tag{3}$$

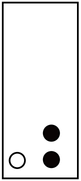
- Dirichlet prior smoothing

$$\pi_{dir} = \frac{\mu}{\mu + |d|} \tag{4}$$

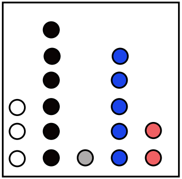
Smoothing II



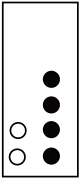
Background Model



Document 1 (Sample)

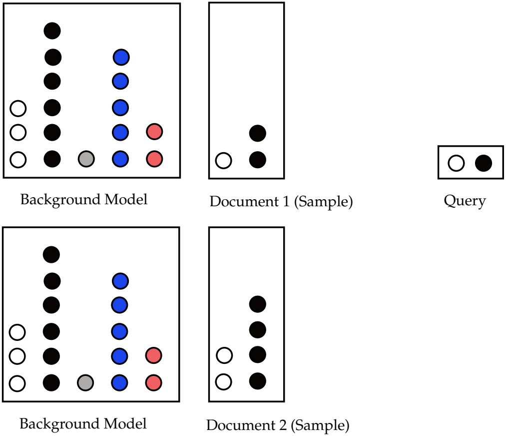


Background Model

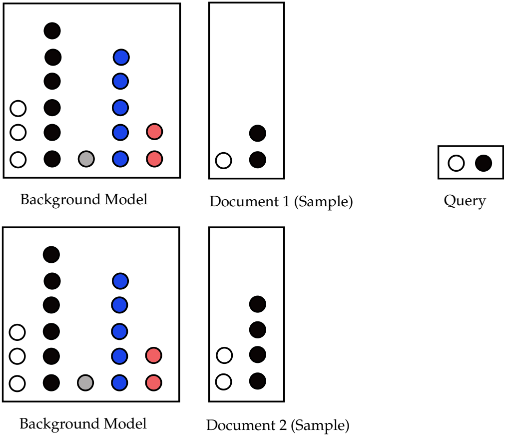


Document 2 (Sample)

Smoothing II



Smoothing II



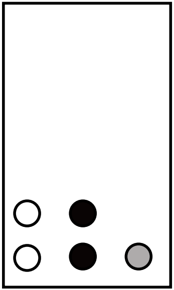
Rank document 2 higher than document 1

Overview

- We can derive a retrieval function (and principled term-weights) using language models, unlike the VSM
- It can be viewed as a form of unsupervised machine learning
- The multinomial model is efficient to estimate and with Dirichlet priors smoothing is the state-of-the-art in terms of retrieval effectiveness
- It forms the basis of many applications
- It does not model term-dependencies
- The model using a Dirichlet prior has a free parameter (i.e. μ)

Outline

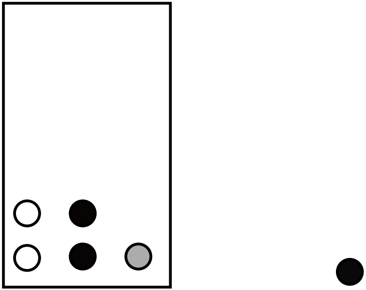
Multivariate Pólya Urn



Model (urn)

Document (sample)

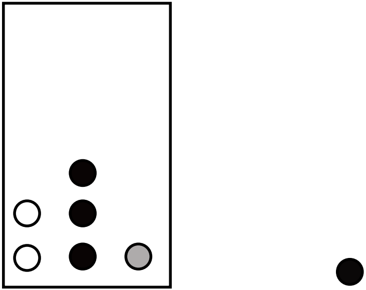
Multivariate Pólya Urn



Model (urn)

Document (sample)

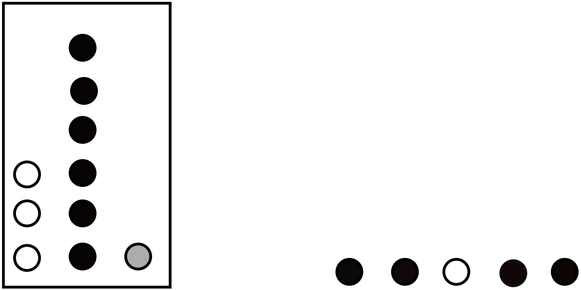
Multivariate Pólya Urn



Model (urn)

Document (sample)

Multivariate Pólya Urn



Model (urn) Document (sample)

Sampling with reinforcement

the rich get richer

Multivariate Pólya Distribution

- The multivariate Polya distribution (Dirichlet-compound-multinomial or DCM)
- Instead of the multinomial in the original query-likelihood model we can use the DCM

$$p(d|\alpha) = \int_\theta p(d|\theta)p(\theta|\alpha)d\theta \tag{5}$$

- Parameter vector α can be interpreted as the initial number of balls of each colour in the urn

Parameterisation

$$\alpha_d = m_d \cdot \theta_d = (m_d \cdot p(t_1|\theta_d), m_d \cdot p(t_2|\theta_d), ..., m_d \cdot p(t_v|\theta_d)) \quad (6)$$

- θ_d can be seen as the expectation
- m_d can be seen as the scale (variance)
- Low m_d implies high burstiness

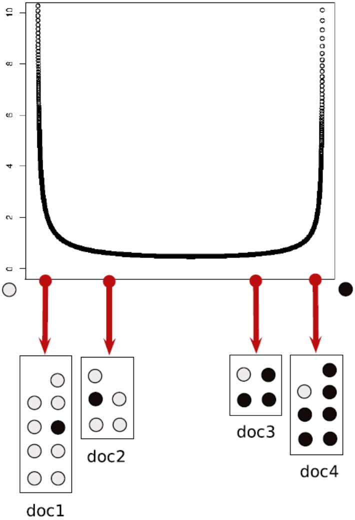
Some properties

- Subsequent balls drawn from the urn are identically distributed but dependent
- Each sample (document) can be modelled using a multinomial
- Each time you *restart* the process to draw a sample, you draw from a different multinomial
- The process is *exchangable*
- Generates power-law characteristics of term-frequencies
- Estimating a DCM from multiple samples (i.e. multiple documents) is computationally expensive (i.e. no closed-form solutions)

The SPUD Language Model

- Use the Pólya urn as a model for document generation
- Documents are known to exhibit burstiness
- Estimate the document and background models as before but with different model assumptions
- Retain the multinomial as the model for query generation

Background Model I



Background Model II

- The background model is the most likely single model to have generated all documents
- Given all documents, find the DCM parameters
- Elkan (2006) has shown that close approximations to the model parameters are proportional to the number of samples in which an observation appears (EDCM)
- Essentially, documents exhibit quite a lot of word burstiness

Background Model II

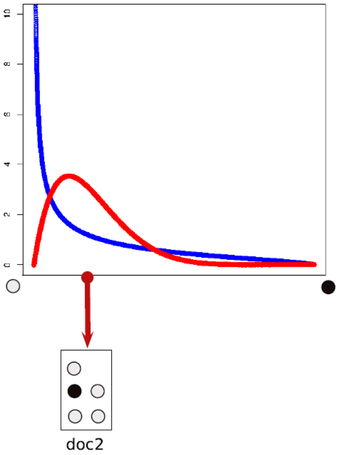
- This is a useful result as close estimates of the the background parameters will be proportional to:

$$p(t|\hat{\theta}'_c) = \frac{df_t}{\sum_{t'} df_{t'}} = \frac{df_t}{\sum_j^n |\vec{d_j}|} \tag{7}$$

- With only m_c remaining to be estimated using Newton's method

$$\hat{\alpha}_c = (m_c \cdot p(t_1|\hat{\theta}'_c), m_c \cdot p(t_2|\hat{\theta}'_c), ..., m_c \cdot p(t_v|\hat{\theta}'_c)) \tag{8}$$

Document Model I



Document Model II

- With only one sample we cannot estimate the parameters of a DCM
- We can estimate the expectation of the DCM but what is m_d?
- Thought experiment: What is the minimum initial mass of the urn (i.e. number of balls) that could have generated d?

Document Model II

- With only one sample we cannot estimate the parameters of a DCM
- We can estimate the expectation of the DCM but what is m_d?
- Thought experiment: What is the minimum initial mass of the urn (i.e. number of balls) that could have generated d?
- We set m_d to the number unique terms in the document (it’s lower bound).

Document Model II

- With only one sample we cannot estimate the parameters of a DCM
- We can estimate the expectation of the DCM but what is m_d?
- Thought experiment: What is the minimum initial mass of the urn (i.e. number of balls) that could have generated d?
- We set m_d to the number unique terms in the document (it's lower bound).

$$\hat{\alpha}_d = (|\vec{d}| \cdot p(t_1|\hat{\theta}_d), |\vec{d}| \cdot p(t_2|\hat{\theta}_d), ..., |\vec{d}| \cdot p(t_v|\hat{\theta}_d)) \qquad (9)$$

Remaining Parameters

- Linearly combine the two models using one parameter ω
- We can experimentally tune ω

$$SPUD = \omega \cdot \alpha_c + (1 - \omega) \cdot \alpha_d \tag{10}$$

Remaining Parameters

- Linearly combine the two models using one parameter ω
- We can experimentally tune ω

$$SPUD = \omega \cdot \alpha_c + (1 - \omega) \cdot \alpha_d \tag{10}$$

$$\hat{\alpha}_c = (m_c \cdot p(t_1|\hat{\theta}'_c), m_c \cdot p(t_2|\hat{\theta}'_c), ..., m_c \cdot p(t_v|\hat{\theta}'_c)) \tag{11}$$

$$\hat{\alpha}_d = (|\vec{d}| \cdot p(t_1|\hat{\theta}_d), |\vec{d}| \cdot p(t_2|\hat{\theta}_d), ..., |\vec{d}| \cdot p(t_v|\hat{\theta}_d)) \tag{12}$$

Outline

Questions

- How effective is the new model in terms of retrieval?
- How effective is Newton's method at automatically determining the free parameter in the background model?
- Why?

Effectiveness MAP

- Optimally tuning the one free parameter in each function
- All increases are statistically significant (for SPUD v MQL)

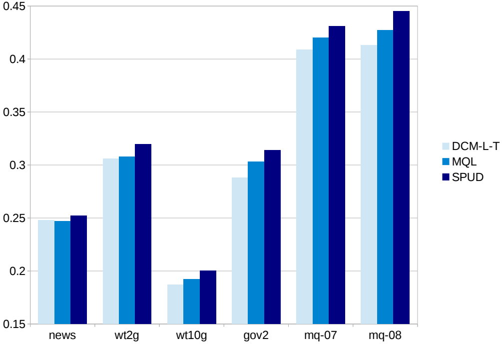


Figure : MAP on Newswire and Web datasets (title only queries)

Effectiveness NDCG@20

- Optimally tuning the one free parameter in each function
- All increases are statistically significant

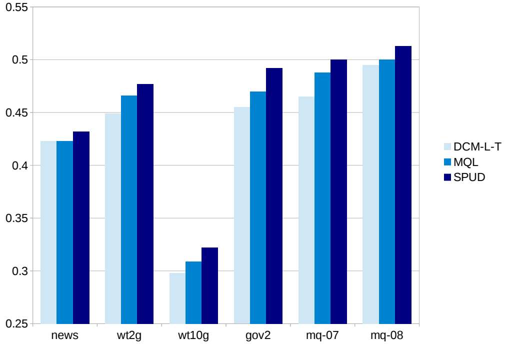


Figure : NDCG@20 on Newswire and Web datasets (title only queries)

Newton's Method and Tuning

- Mixing parameter is robust at $\omega = 0.8$

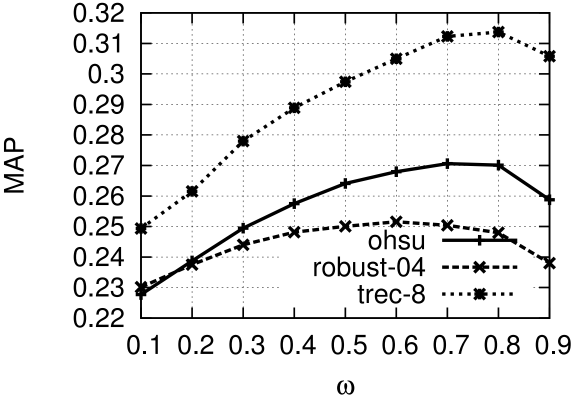


Figure : (title queries)

Newton's Method

- Mixing parameter is set to $\omega = 0.8$
- Tuned m_c vs m_c estimated using Newton's method

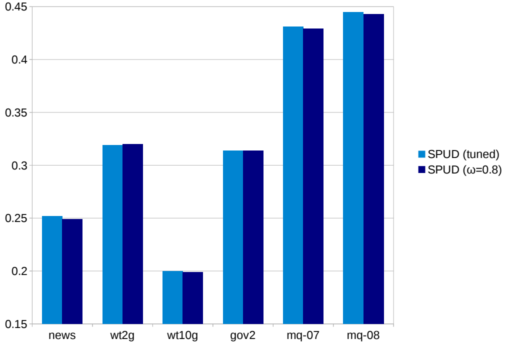


Figure : MAP on Newswire and Web datasets (title only queries)

Outline

Scope Hypothesis

- One of two hypotheses proposed that aim to explain the interaction between document length and topicality
- Documents vary in length due to some documents covering more topics (Robertson & Walker, 1994)
- Relevance is likely affected by this aspect of document length

Verbosity Hypothesis

- Documents vary in length due to verbosity (Robertson & Walker, 1994)
- Some documents are just more ‘wordy’
- This aspect of document length is independent of topic, and therefore, relevance
- In reality documents may vary in length due to a combination of these two hypotheses
- No formal means of capturing whether a retrieval function adheres to this intuition has been proposed (as far as I know)

Axiomatic Analysis

LNC2* Constraint

Given a ranking function $s(q, d)$ that scores a document d with respect to a query q, if d' is created by concatenating d with itself k times, then $s(q, d') = s(q, d)$

Axiomatic Analysis

LNC2* Constraint

Given a ranking function $s(q, d)$ that scores a document d with respect to a query q, if d' is created by concatenating d with itself k times, then $s(q, d') = s(q, d)$

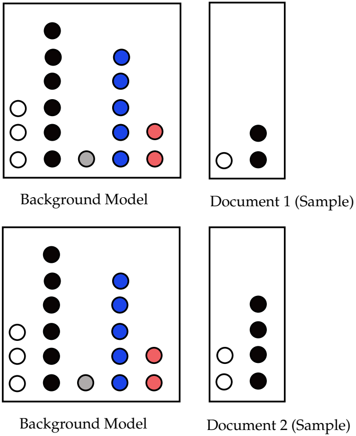
In other words, you cannot change the ranking of a document by concatenating it with itself k times (where $k > 1$)

Comparison

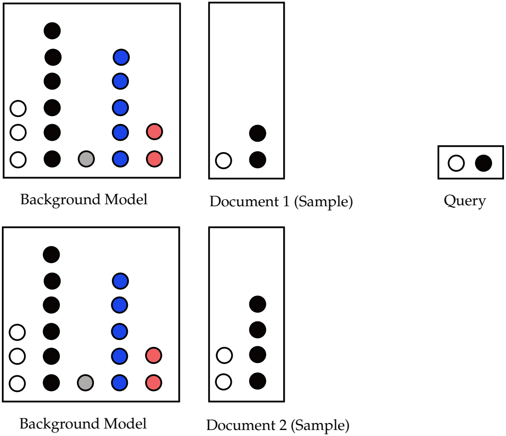
Multinomial

$$\mathrm{MULT}_{dir}(q,d) = \sum_{t \in q} log\left(\frac{|d|}{|d|+\mu} \cdot \frac{c(t,d)}{|d|} + \frac{\mu}{|d|+\mu} \cdot \frac{cf_t}{|C|}\right) \cdot c(t,q) \tag{13}$$

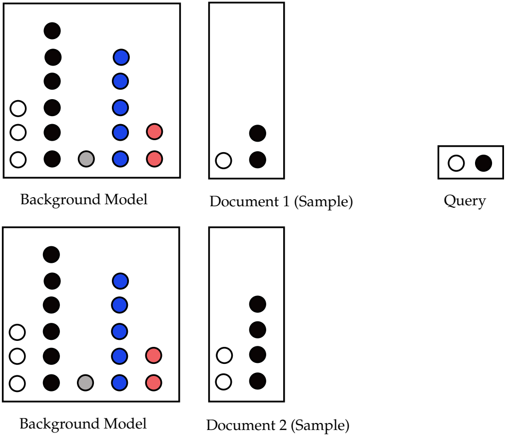
Violation I


Background Model
Document 1 (Sample)
Background Model
Document 2 (Sample)

Violation I

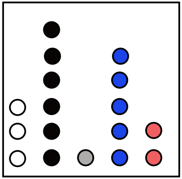

Background Model
Document 1 (Sample)
Query
Background Model
Document 2 (Sample)

Violation I

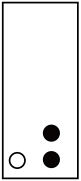


Document 2 is ranked higher than document 1 (X)

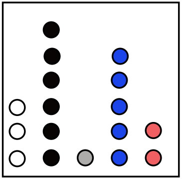
Violation II



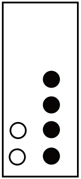
Background Model



Document 1 (Sample)

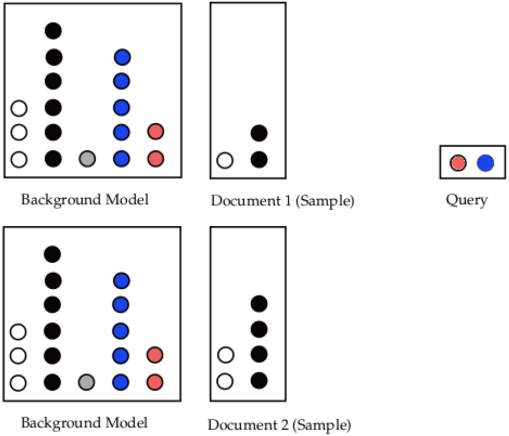


Background Model

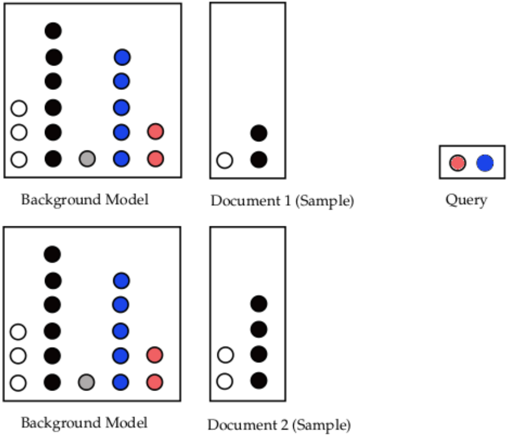


Document 2 (Sample)

Violation II



Violation II



Document 1 is ranked higher than document 2 (X)

Comparison

Multinomial

$$\mathrm{MULT}_{dir}(q,d) = \sum_{t \in q} log\left(\frac{|d|}{|d|+\mu} \cdot \frac{c(t,d)}{|d|} + \frac{\mu}{|d|+\mu} \cdot \frac{cf_t}{|C|}\right) \cdot c(t,q) \tag{14}$$

Comparison

Multinomial

$$\mathrm{MULT}_{dir}(q,d) = \sum_{t \in q} log(\frac{|d|}{|d|+\mu} \cdot \frac{c(t,d)}{|d|} + \frac{\mu}{|d|+\mu} \cdot \frac{cf_t}{|C|}) \cdot c(t,q) \tag{14}$$

SPUD

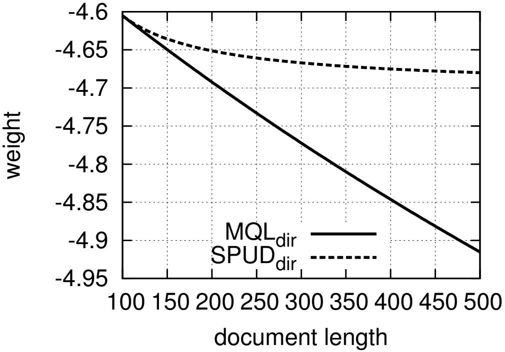
$$\mathrm{SPUD}_{dir}(q,d) = \sum_{t \in q} log(\frac{|\vec{d}|}{|\vec{d}|+\mu'} \cdot \frac{c(t,d)}{|d|} + \frac{\mu'}{|\vec{d}|+\mu'} \cdot \frac{df_t}{\sum_t df_t}) \cdot c(t,q) \tag{15}$$

What about scope?

- What happens as non-query (off-topic) terms are added to a document
- The part of the SPUD model that deals with scope, only penalises documents as distinct terms are added

What about scope?

- What happens as non-query (off-topic) terms are added to a document
- The part of the SPUD model that deals with scope, only penalises documents as distinct terms are added



Probability of Relevance/Retrieval

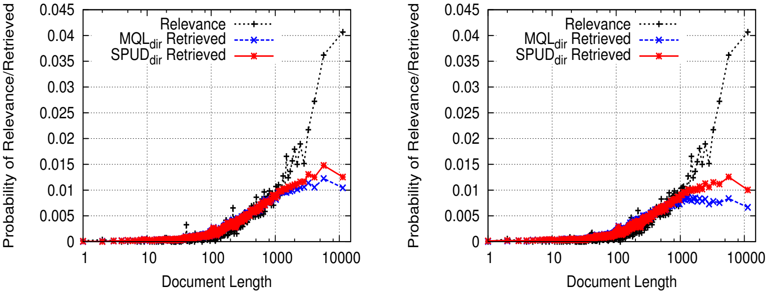


Figure : Probability of retrieval/relevance for MQL$_{dir}$ and SPUD$_{dir}$ methods for trec-9/01 collection for short queries (left) and medium length queries (right).

A more sensitive idf

Traditional idf

$$log(\frac{n}{df_t}) \tag{16}$$

A more sensitive idf

Traditional idf

$$log(\frac{n}{df_t}) \tag{16}$$

The actual weight applied to a term occurring in a document can be re-written as follows:

variable idf

$$log(1 + \delta \cdot \frac{n}{df_t}) \tag{17}$$

where $\delta = c(t, d) \cdot |\vec{d}|_{avg} \cdot |\vec{d}|/(\mu' \cdot |d|)$ contains term-frequency and document length normalisation

Outline

Conclusions

- The simple bag-of-words approach has not yet reached its limit
- More accurately modelling the language generation process leads to more accurate unsupervised models of retrieval
- The Pólya urn model leads to more effective retrieval without any additional cost
- Automatic setting the parameters in the background model (unlike the multinomial model)
- An analysis shows that the new model adheres to a new test for the verbosity hypothesis

Acknowledgements and Questions

- Jiaul Paik (University of Maryland)
  and
  Yuanhua Lv (Microsoft Research)

Acknowledgements and Questions

- Jiaul Paik (University of Maryland)
  and
  Yuanhua Lv (Microsoft Research)

- Questions and comments welcome

Church, K. W. and Gale, W. A. (1995).
Poisson mixtures.
Natural Language Engineering, **1**, 163–190.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2011).
Producing power-law distributions and damping word frequencies with two-stage language models.
Journal of Machine Learning Research, **12**, 2335–2382.

Hiemstra, D. (1998).
A linguistically motivated probabilistic model of information retrieval.
In *Research and Advanced Technology for Digital Libraries, Second European Conference*, ECDL '98, pages 569–584.

Lavrenko, V. and Croft, W. B. (2001).
Relevance based language models.
In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA. ACM.

Lv, Y. and Zhai, C. (2009).
Positional language models for information retrieval.
In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 299–306, New York, NY, USA. ACM.

Madsen, R. E., Kauchak, D., and Elkan, C. (2005).
Modeling word burstiness using the dirichlet distribution.
In *Proceedings of the 22nd International Conference on Machine Learning*, pages 545–552.

Mitzenmacher, M. (2003).
A brief history of generative models for power law and lognormal distributions.
INTERNET MATHEMATICS, **1**, 226–251.

Ponte, J. M. and Croft, W. B. (1998).
A language modeling approach to information retrieval.
In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information*

Retrieval, SIGIR '98, pages 275–281, New York, NY, USA. ACM.

Robertson, S. E. and Walker, S. (1994).
Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval.
In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 232–241, New York, NY, USA. Springer-Verlag New York, Inc.

Simon, H. A. (1955).
On a class of skew distribution functions.
Biometrika, **42**(3–4), 425–440.

Xu, Z. and Akella, R. (2008).
A new probabilistic retrieval model based on the dirichlet compound multinomial distribution.
In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information*

Retrieval, SIGIR '08, pages 427–434, New York, NY, USA. ACM.

Zhai, C. and Lafferty, J. (2001).
A study of smoothing methods for language models applied to ad hoc information retrieval.
In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA. ACM.

Zhai, C. and Lafferty, J. (2004).
A study of smoothing methods for language models applied to information retrieval.
ACM Transactions of Information Systems, **22**, 179–214.