Statistical Language Modeling for Information Access

Theory, day 1: Basics and practicalities

Maarten de Rijke   Edgar Meij   Kristian Balog

University of Amsterdam
Norwegian University of Science and Technology

August 1–4, 2011

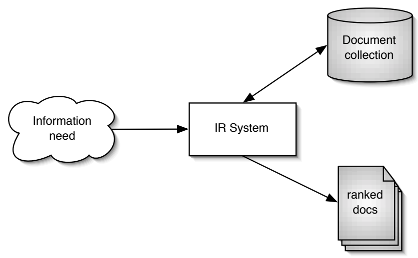
Outline

Search

- Information avalanche
  - Internet
  - Intelligence
  - Scientific research (astronomy, biomedicine, humanities, . . .)
  - Cultural heritage
  - Desktop, Email, . . .
  - Enterprise, Business Intelligence
  - User generated content
  - . . .
- Not just growing, but growing at a growing pace
  - 1999: 250 megabytes per person for each man, woman, and child on earth
  - 2002: almost 800 MB of recorded information is produced per person
  - `http://www.sims.berkeley.edu/research/projects/how-much-info-2003/`
  - Today?

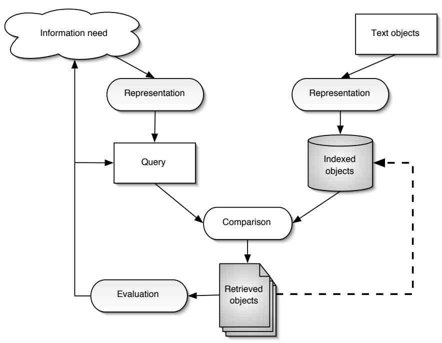
Thought Experiment

- Imagine that your are an information retrieval engine



- What do you do?

Thought Experiment (2)



Basic Information Retrieval

- Given an information need, return suitable results
  - Document retrieval: Given a free text query, produce a list of documents ranked from most to least relevant
  - "Relevant" ∼ "about the same topic"
  - "About the same topic" ∼ "similarity"
- Basic idea at the heart of much work in IR
  - find words in docs
  - compare them to words in query
  - some words get a bigger weight than others
  - this approach is extremely effective!

Basic Information Retrieval

- **Bag of words** representation of contents of documents
  - effective and popular approach, considers words without order or structure
  - look at all re-arrangments of newspaper headline
    - stocks fall on inflation fears
    - inflation stocks fall on fears
    - fall inflation stocks on fears
    - fall fears inflation stocks on
    - fall fears inflation on stocks
- IR research builds on basic idea of comparing bags of words
  - what is the value/weight of a word?
  - how do we determine similarity?
  - can we get a formal/theoretical model for this?

The Meaning of "Meaning"

- Meaning = use . . .
  - Observe language used in query
  - Observe language used in documents
  - Compare these observations
  - Count, count, count, . . .
- Other features used in query-document comparisons
  - Phrases
  - Link structure
  - Named entities (people, locations, times, organizations, products, . . .)
  - . . .
- Research into effectiveness, efficiency, and extending the ideas to new settings

Language Modeling for Information Access

- Intuition
  - Users
    - Have a reasonable idea of terms that are likely to occur in documents of interest
    - Will choose query terms that distinguish these documents from others in the collection
- Language modeling approaches
  - Attempt to model query generation process
    - Different estimation methods, (in)dependence assumptions, ...
  - Documents are ranked by probability that query would be observed as a random sample from the respective document model
    - Suitable variations for other retrieval tasks

IR Methodology

- But does it work?
- IR has a very heavy emphasis on experimental evaluation
  - Often comparative: given System A and System B, use a suitable test collection to score both, then analyze the differences (if any)
- Theory meets Experiment meets Practice
  - Real World Task™
    - → suitably abstracted into test collection
    - → devise, compare, improve models and algorithms
  - Test collection development often done as collaborative effort
  - Increasing awareness of need to supplement lab-based evaluations with user studies: it works, but do users become happier?

Outline

Outline of the Course

Theory

- The course wiki
  - http://www.science.uva.nl/~mdr/Teaching/Cordoba2011/
  - SouthWestOfAmsterdam (case sensitive!)
- Day 1: general retrieval modeling and evaluation principles; introduction to language modeling
- Day 2: estimation, smoothing methods, mixture models, and applications to retrieving (semi)structured documents
- Day 3: incorporating symbolic knowledge, lexical relations and context within a language modeling setting
- Day 4: language modeling approaches to tasks at the interface of IR and IE; ongoing developments and prominent research questions

Practical Component

- Aim: basic familiarity with Lemur
  - Language modeling toolkit developed at UMass
  - `http://lemurproject.org/tutorials/`
  - `http://ciir.cs.umass.edu/~strohman/indri`
- Higher aim: you should be able to run an information retrieval experiment using Lemur by the end of the week
  - Index, submit queries, generate results, evaluate the results, compare and analyse the outcomes, ...

Outline of the Course

Practical

- Day 1: Installing and Indexing
- Day 2: Retrieval and Evaluation
- Day 3: Retrieval Parameters
- Day 4: Pseudo Relevance Feedback; Additional bells, whistles and requests

Learning Goals

Things we want to get across

- Basic information retrieval, including evaluation methodology
- Basic language modeling for IR, applications of language modeling ideas to a broad range of information access tasks
- A sense of today's state of the art in language modeling in IR
- Hands-on experience with the Lemur, language modeling toolkit
- Familiarity with the basic "experimental loop" in IR

Who Are We?

- Maarten de Rijke
  - Worked in modal logic for 10 years, then switched to IR
  - Currently professor of "Information processing and Internet," leading an IR group of about 25 people (ILPS)
  - Main interests: intelligent information access, social media analysis, beyond relevance, beyond the ranked list, learning to rank
- Edgar Meij
  - Postdoc in said group
  - Main interests: Leveraging conceptual knowledge from (structured) knowledge source to enhance information access
- Krisztian Balog
  - Former postdoc in said group
  - Main interest: Entity related search, semantic search, evaluation

Outline

1. **Introduction**
   Background
2. **A look ahead**
3. **Let's get to work**
   Basic language modeling
   Basic evaluation

Some Elementary Material

Some Elementary Material

- Assume basic familiarity with statistics ("you can count")
- Bayes:

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

- Maximum likelihood estimation: method used for fitting a mathematical model to some data; a way of tuning the free parameters of the model to provide a good fit
- Elementary notions about graphs
- Less than a tiny bit of XML, HTML
- Theory meets experiments meets application
- Search experience

Some Elementary Material

- Term frequency, (inverse) doc frequency, doc length normalization
- Term frequency (TF): frequency of word w in document d

$$tf_{w,d} = \frac{word\_count(w,d)}{word\_count(d)}$$

- Inverse document frequency (IDF):

$$\begin{aligned} df_w &= \text{number of docs in which } w \text{ appears} \\ idf_{w,d} &= \log\left(\frac{\text{number of docs}}{df_w}\right) \end{aligned}$$

- Weight of term w in doc d

$$weight_{w,d} = tf_{w,d} \cdot idf_{w,d}$$

- Baseline vector-based similarity

$$sim(q,d) = \frac{\sum_{w \text{ in } q} weight_{w,d} \cdot weight_{w,d}}{\sqrt{\sum_{w \text{ in } d} weight^2_{w,d}} \cdot \sqrt{\sum_{w \text{ in } q} weight^2_{w,q}}}$$

Retrieval Based on Language Models

- Treat the generation of queries as a random process
- Approach
  - Infer a language model for each document.
  - Estimate the probability of generating the query according to each of these models.
  - Rank the documents according to these probabilities.
  - Usually a unigram estimate of words is used
- What's a language model? Probability distribution over strings
  - how likely is a given string (observation) in a given "language"
  - English: $p_1 > p_2 > p_3 > p_4$
    - $p_1 = P($"a quick brown fox"$)$
    - $p_2 = P($"fox a quick brown"$)$
    - $p_3 = P($"een snelle brown fox"$)$
    - $p_4 = P($"een snelle bruine vos"$)$

What's a Language Model?

- … depends on what "language" we are modeling
  - in much of IR $p_1 = p_2$
  - in some applications we may want p_3 to be high
- Notation
  - Convention: make explicit what we are modeling
    - M: "language" we are trying to model
    - s: observation (string of tokens from some vocabulary)
    - $P(s|M)$: probability of observing "s" in language M
  - What is M?
    - a "source" or "generator": a mechanism that spits out strings that are legal in the language
    - $P(s|M)$: probability of getting "s" during random sampling from M

Language Modeling for IR

- Every document in a collection defines a "language"
  - consider all posssible sentences (strings) that author could have written down when creating some given document
  - some are perhaps more likely to occur than others
    - … subject to topic, writing style, language, …
  - $P(s|M_D)$: probability that author would write down string "s"
    - think of writing zillions of variations of a document and counting how many times we get "s"
- Suppose q is the user's query
  - what is the probability that author would write down "q"?
- Rank documents D in the collection by $P(q|M_D)$
  - probability of observing q during random sampling from the language model of document D

Other Apps: Same Idea

- Topic detection and tracking
  - query q can be topic description, or an on-topic story
  - documents with high $P(q|M_D)$ probably discuss the same topic
- Classification/filtering
  - query can be a set of training documents for a particular class
  - or testing docs can refelct observations from model of training set
- Cross-language retrieval
  - query can be in a different language from document collection
  - author could have written a document in a different language
- Multi-media retrieval
  - languages don't have to be textual (e.g., spoken or handwritten docs)
  - extends to images, sounds, video, preferences, hyerplinks, …
- Expert finding
  - ?

Unigram LMs

- Words are sampled independently from each other
  - metaphor: randomly pulling out words from an urn (with replacement)
  - joint probability decomposes into a product of marginals
  - estimation of probabilities: simple counting
- E.g., assume $M = \{R, B, R, B, Y, B, R, R, Y\}$ and q, the query, is $\{R, Y, R, B\}$
  - $P(q) = P(R) \cdot P(Y) \cdot P(R) \cdot P(B) = 4/9 \cdot 2/9 \cdot 4/9 \cdot 3/9$

Ranking with LMs

- Standard approach: **query likelihood**
  - estimate a language model M_D for every document D in the collection
  - rank docs by the probability of "generating" the query

$$P(q_1, \ldots, q_k | M_D) = \prod_{i=1}^{k} P(q_i | M_D)$$

- Drawbacks
  - no notion of relevance in the model: everything is random sampling
  - user feedback/query expansion not part of the model
    - examples of relevant documents cannot help us improve the language model M_D
    - the only option is augmenting the original query Q with extra terms
    - we could, in principle, make use of sample queries for which D is relevant
  - does not directly allow weighted or structured queries

Estimation

- Want: estimate M_Q and/or M_D from Q and/or D
- General problem
  - given a string of text S (Q or D), estimate its language model M_S
  - S is commonly assumed to be (independent and identically distributed) random sample from M_S
- Basic LMs
  - maximum likelihood estimator and the zero frequency problem
  - discounting techniques
    - Laplace correction, Lindstone correction, absolute discounting, leave-one-out discounting, Good-Turing method
  - interpolation/back-off techniques
    - Jelinek-Mercer smoothing, Dirichlet ssmoothing, Witten-Bell smoothing, Zhai-Lafferty two-stage smoothing, interpolation vs. back-off techniques
  - Bayesian estimation

Maximum-Likelihood

- Count relative frequencies of words in S
  - $P_{mle}(w|M_S) = \#(w, S)/|S|$
  - if $S = \{B, R, Y\}$, we get $P(B) = P(R) = P(Y) = 1/3$ and $P(W) = P(G) = 0$
- Maximum-likelihood property
  - assigns highest possible likelihood to the observation
- Unbiased estimator
  - if we repeat estimation an infinite number of times with different starting points S, we will get correct probabilities (on average)
  - somewhat problematic to operationalize...

Zero-Frequency Problem

- Suppose some event not in our observation S
  - model will assign zero probability to that event
  - and to any set of events involving the unseen event
- Happens very frequently with language → Zipf
- It is incorrect to infer zero probabilities
  - especially when creating a model from short samples
- If $S = \{B, R, Y\}$, what is $P(RYGBRYBRYB)$?

Discounting Methods

- Laplace correction
  - add 1 to every count, normalize
  - problematic for large vocabularies
  - add a small constant ϵ to every count, re-normalize
- Absolute discounting
  - substract a constant ϵ, re-distribute the probability mass
- Example: $S = \{B, R, Y\}$ "$+\epsilon$"
  - $P(B) = P(R) = P(Y) = (1 + \epsilon)/(3 + 5\epsilon)$
  - $P(G) = P(W) = (0 + \epsilon)/(3 + 5\epsilon)$

Interpolation Methods

- Problem with all discounting methods
  - discounting treats unseen words equally (add or subtract ϵ)
  - some words are more frequent than others
- Idea: use background probabilities
  - "interpolate" maximum likelihood estimates with, e.g., General English expectations (computed as relative frequency of a word in a large collection)
  - reflects expected frequency of events
  - in IR applications, plays the role of IDF
- 2-state HMM analogy
  - $\lambda \cdot S + (1 - \lambda)GE$

Jelinek-Mercer Smoothing

- Correctly setting λ is very important
- Start simple
  - set λ to be a constant, independent of document, query
- Tune to optimize retrieval performance
  - optimal value of λ varies with different text collections, tasks, query sets, evaluation metrics, etc.

Basic LM Approach: Summary

- Goal: estimate a model M from a sample text S
- Use maximum likelihood estimator
  - count the number of times each word occurs in S, divide by length
- Smoothing to avoid zero frequencies
  - discounting methods: add or subtract a constant, redistribute mass
  - better: interpolate with background probability of a word
  - smoothing has a role similar to IDF in classical models
- Smoothing parameters very important
  - Dirichlet works well for short queries (need to tune parameter)
  - Jelinek-Mercer works well for longer queries (also needs tuning)

Things to Think About

- Text representation
  - What makes a "good" representation?
  - How is a representation generated from text?
  - What are retrievable objects and how are they organized?
- Representing information needs
  - What is an appropriate query language?
  - How can interactive query formulation and refinement be supported?
- Comparing representations
  - What is a "good" model of retrieval?
  - How is uncertainty represented?

What's It All About . . . Relevance

- Relevance is difficult to define satisfactorily
- A relevant document is one judged useful in the context of a query
  - who judges? what is useful?
  - humans not very consistent
  - judgments depend on more than document and query
- With real collections, never know all relevant docs
- Assessing retrieval: boring and *very* time-consuming

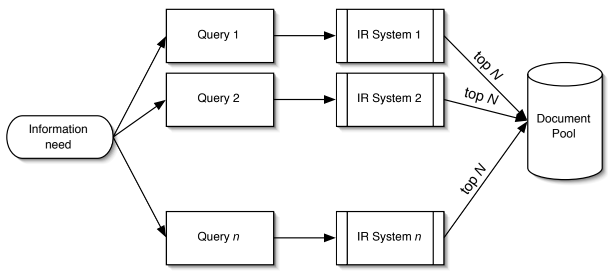
Test Collections

- Compare retrieval performance using a test collection
  - set of documents
  - set of queries
  - set of relevance judgments (which docs relevant to each query)
- To compare the performance of two techniques:
  - each technique used to evaluate test queries
  - results (set or ranked list) compared using performance measure
  - most common measures based on precision and recall
- Use multiple measures to get different views
  - test with multiple collections — performance is collection dependent

Finding Relevant Documents

- Question: did the system find **all** relevant material?
- To answer accurately, corpus needs complete judgments
  - i.e., "yes," "no," or some score for every query-document pair
- For small corpora, can review all docs for all queries
  - done for TDT collection of 60K docs as recently as 1998
- Not practical for large or medium-sized collections
  - TREC collections have millions of documents
- Other approaches that can be used
  - sampling, search-based, pooling

Finding Relevant Documents: Pooling



Precision and Recall

- Precision: fraction of retrieved documents that is relevant

$$precision = \frac{|relevant \cap retrieved|}{|retrieved|}$$

- Recall: fraction of the relevant documents that has been retrieved

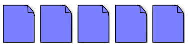
$$precision = \frac{|relevant \cap retrieved|}{|relevant|}$$

- All relevant docs in the collection: A B C D
  Retrieved docs: A C D E F
- $P = 3/5, R = 3/4$

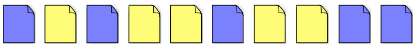
Precision and Recall

- P and R are well-defined for sets
- For ranked retrieval. . .
  - compute a P/R point for each relevant document
  - compute a value at fixed recall points (e.g., precision at 20% recall)
  - compute value at fixed rank cutoffs (e.g., precision at rank 20)
  - . . .

Precision and Recall Example

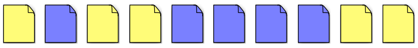
Five relevant documents: 

- Ranking #1:



Recall	**0.2**	0.2	**0.4**	0.4	0.4	**0.6**	0.6	0.6	**0.8**	**1.0**
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

- Ranking #2:



Recall	0.0	**0.2**	0.2	0.2	**0.4**	**0.6**	**0.8**	**1.0**	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.63	0.55	0.5

Average Precision

- Often want a single-number effectiveness measure
  - e.g., for a machine learning algorithm to detect improvements
- **Average precision** is widely used
  - calculate by averaging precision when recall increases

Recall	**0.2**	0.2	**0.4**	0.4	0.4	**0.6**	0.6	0.6	**0.8**	**1.0**
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

Average precision $= \mathbf{(1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62}$

Recall	0.0	**0.2**	0.2	0.2	**0.4**	**0.6**	**0.8**	**1.0**	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.63	0.55	0.5

Average precision $= \mathbf{(0.5 + 0.4 + 0.5 + 0.57 + 0.63)/5 = 0.52}$

Averaging

- Hard to compare P/R scores for individual queries
  - need to average over many queries
- Two main types of averaging
  - micro-average – each relevant doc is a point in the average
  - macro-average – each query is a point in the average (most common)
  - what does each tell someone evaluating a system?
- Why use one over the other?
- Also done with average precision value
  - called mean average precision (MAP)

Average Precision Again

- Average precision at standard recall points
- For a given query, compute P/R point for every relevant doc.
- Interpolate precision at standard recall levels
  - 11-pt is usually 100%, 90, 80, , 10, 0% (yes, 0% recall)
  - 3-pt is usually 75%, 50%, 25%
- Average over all queries to get average precision at each recall level
  - average over all recall levels to get a single result
  - called "interpolated average precision"

Some Other Single-Valued Measures

- F measure
  - $F = 1 - E$ often used (good results mean larger values of F)
  - $F1$ measure is popular: F with $\beta = 1$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$$F_1 = \frac{1}{1/P + 1/R}$$

- R-Precision
  - given a query Q compute the precision at rank $|relevant_Q|$, where $relevant_Q$ is the set of relevant docs for Q
- $p@n$
  - compute the precision at a fixed rank n for every query
  - useful for evaluating search engines

Known Item Search

- Site finding
  - a **site** is an organized collection of pages on a specific topic maintained by a single person or group
  - not the same thing as a domain (`cnn.com` has numerous sites)
- Topic can be very broad; examples from a query log:
  - *Where can I find Hotmail?*
  - *Where is the official Star Wars site?*
  - *Where is the fun site dating patterns analyzer?*
- Not known-item, but known answer (question answering)
  - *Who was Cleopatra?* or: *Where is the Taj Mahal?*
- Given a query, find **the** URL or **the** answer

Evaluating Known Item

- Usually only one possible answer (the site's page)
  - so recall is either zero or one
  - recall/precision graphs are not very interesting
- Instead measure the rank where the site's URL was found
  - sometimes scored as $1/rank$
  - when averaged over many queries, called "mean reciprocal rank" (MRR)

Significance Tests

- Are observed differences statistically different?
  - generally can't make assumptions about underlying distribution
  - single-valued measures are easier to use, but R/P is possible
- Sign test or Wilcoxon signed-ranks test are typical
  - sign test answers how often
  - Wilcoxon answers how much
- Bootstrapping methods
- Are observed differences detectable by users?

Sign Test Example

- For techniques A and B, compare average precision for each pair of results generated by queries in test collection
  - if difference is large enough, count as $+$ or $-$, otherwise ignore
  - use number of $+$'s and the number of significant differences to determine significance level
    - e.g. for 40 queries, technique A produced a better result than B 12 times, B was better than A 3 times, and 25 were the "same", $p < 0.035$ and technique A is significantly better than B
    - if $A > B$ 18 times and $B > A$ 9 times, $p < 0.122$ and A is not significantly better than B at the 5% level

Retrieval Evaluation with Incomplete Information

- Buckley and Voorhees, SIGIR 2004
  - Is the Cranfield evaluation methodology robust to gross violations of the completeness assumption?
  - I.e., what if the assumption that all relevant documents within a test collection have been identified and are present in the collection is incorrect?
- Current evaluation measures not robust to substantially incomplete relevance judgments
  - e.g., p@10 is a lot less robust that avg. precision
- New measure introduced
  - highly correlated with existing measures when complete judgments are available
  - more robust to incomplete judgment sets
- Lots of ongoing research as collection sizes grow and sets of known relevant items become grossly incomplete

Wrap Up and Look Ahead

- The course wiki
  - `http://www.science.uva.nl/~mdr/Teaching/Cordoba2011/`
  - SouthWestOfAmsterdam (case sensitive!)
- Summary
  - A bit on information retrieval
  - Basic language modeling for IR, with a bit on smoothing
  - Basic evaluation methodology: precision, recall, mean average precision
- Tomorrow
  - A bit more on evaluation
  - More on estimation, smoothing, mixture models, priors
  - Applications to retrieving (semi)structured documents, web retrieval
- On to today's practical part