Query Evolution

W. Bruce Croft
University of Massachusetts Amherst

Overview

- What is query evolution?
- Evolution as transformation
- Models of transformation
- Transforming long queries

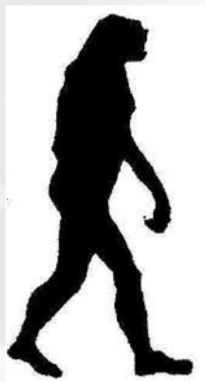
Senses of Evolution

- Evolution in query format since the 70's
- Evolution in the information need over a session
- Evolution in the sense of transforming an initial query into more effective variants
  - focus of this talk

1970 CE
(Boolean search)

NEGLECT! FAIL! NEGLIG! /5 MAINT! REPAIR! /P NAVIGAT! /5 AID EQUIP! LIGHT BUOY "CHANNEL MARKER"

1994 CE
(web search)



negligence navigation aids

2005 CE
(CQA)

Are there any cases which discuss negligent maintenance or failure to maintain aids to navigation such as lights, buoys, or channel markers?

Evolution of Query Format

- Intelligent design or natural selection?
  - Format based on system design
    - e.g., Boolean query languages for Boolean retrieval systems
  - Also dependent on human factors
    - e.g., keyword queries easier than Boolean for most users
  - Format also influenced by system capabilities
    - e.g., long natural language queries do not work well with current search engines, but long queries are common in CQA applications

Evolution and Interaction

- Information needs can change during browsing and “sessions”
- Users specify new queries or systems suggest alternatives
- Note difference between changing information need and query refinement
  - Longer queries can provide better starting point for interaction

Query Transformation

- Our focus is on how queries can be *transformed* to equivalent, potentially better, queries
  - Queries into paraphrases or “translations”
  - Long queries into shorter queries
  - Short queries into longer queries
  - Queries in one domain to queries in other domains
  - Unstructured queries into structured queries

Query Transformation

- Spelling correction and stemming are query transformations at the word level
- Query expansion techniques are transformations at the word or query level
- Query segmentation is a transformation that adds structure
- Web queries are transformed by adding structure

Example Galago Web Query

```
#weight(
        0.1 #weight( 0.6 #prior(pagerank) 0.4 #prior(inlinks))
        1.0 #weight(
            0.9 #combine(
                #weight( 1.0 pet.(anchor) 1.0 pet.(title)
                         3.0 pet.(body) 1.0 pet.(heading))
                #weight( 1.0 therapy.(anchor) 1.0 therapy.(title)
                         3.0 therapy.(body) 1.0 therapy.(heading)))
            0.1 #weight(
                1.0 #od:1(pet therapy).(anchor) 1.0 #od:1(pet therapy).(title)
                3.0 #od:1(pet therapy).(body) 1.0 #od:1(pet therapy).(heading))
            0.1 #weight(
                1.0 #uw:8(pet therapy).(anchor) 1.0 #uw:8(pet therapy).(title)
                3.0 #uw:8(pet therapy).(body) 1.0 #uw:8(pet therapy).(heading)))
        )
```

Generated vs. "Found"

- Transformations such as spelling correction, stemming, and expansion *generate* new queries
- Query suggestion often involves *finding* similar queries in CQA archives or query logs
- Not that much difference since archives can be used to train generative models

Why Query Transformation?

- Models of transformation could *unify* many different query processing steps
- Understanding transformations and evaluating them should lead to improved *relevance (or information need) models*
  - easier to improve effectiveness by improving queries than improving document-based retrieval models?
- *Long queries* are important and transformation is crucial for long queries

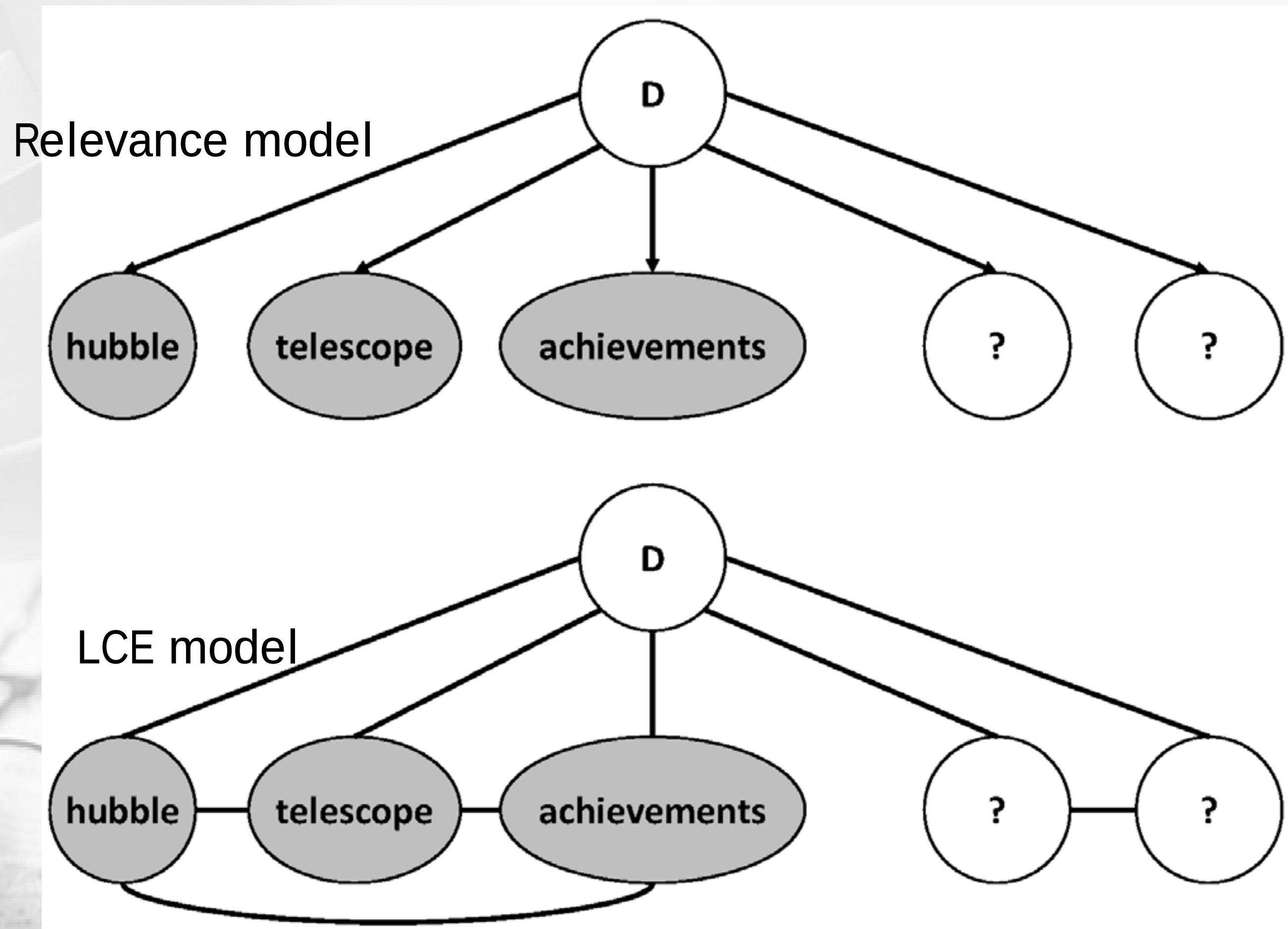
Unifying Query Processing

- Query: *golf curse greens care*
  - spelling correction
    - curse -> course
  - stemming
    - golf -> golfing; curse -> curses, cursing; course -> courses; greens -> green; care -> caring, cared, cares
  - segmentation
    - "golf course", "golf course greens", "greens care"
  - expansion
    - golf course greens care products

Models of Query Transformation

- Relevance model
  - Query expansion modeled using joint probabilities of term occurrence
    - Lavrenko (2009), *A Generative Theory of Relevance.*
- Markov Random Field (MRF) model
  - Weighted linear combination of features, models query dependencies, expansion
    - Metzler and Croft (SIGIR 2005), "A Markov random field model for term dependencies"
    - Metzler and Croft (SIGIR 2007), "Latent concept expansion using Markov random fields"

Pseudo-Relevance Feedback



LCE Example

1-word concepts	*2-word concepts*
telescope	hubble telescope
hubble	space telescope
space	hubble space
mirror	telescope mirror
NASA	telescope hubble
launch	mirror telescope
astronomy	telescope NASA
shuttle	telescope space
test	hubble mirror
new	NASA hubble
discovery	telescope astronomy
time	telescope optical
universe	hubble optical
optical	telescope discovery
light	telescope shuttle

Models of Query Transformation

- Translation models
  - Captures word and phrase substitution
    - e.g., Berger and Lafferty (SIGIR 99), “Information retrieval as statistical translation”
    - e.g., Xue et al (SIGIR 2008), “Retrieval models for question and answer archives”
  - Expansion, paraphrase
    - e.g., Riezler et al (ACL 07), “Statistical machine translation for query expansion in answer retrieval”
  - Word ordering, change in length
    - e.g., Echihabi and Marcu (ACL 2003), “A noisy-channel approach to question answering”

Models of Query Transformation

- Unified models
  - Stemming, spelling correction, segmentation, merging, splitting
    - e.g., Guo et al (SIGIR 2008), "A unified and discriminative model for query refinement"
  - Additions and substitutions
    - e.g., Wang and Zhai (CIKM 2008), "Mining term association patterns from search logs for effective query reformulation"

What About Retrieval Models?

- Query transformation models can generate queries, but how are these queries used in search?
  - Sparck Jones and Tait "Automatic search term variant generation" (1984)
    - Analyzed NL queries into semantic representation and generated new queries
  - Microsoft's QA system AskMSR (2002)
    - Generated queries to retrieve relevant snippets using query rewrite rules

What About Retrieval Models?

- Relevance models and translation models have been associated with retrieval models, but may be too simplistic
  - e.g., unigram model of possible queries
  - e.g., query viewed as a "translation" of a document?
- MRF is a possible framework to represent many features of queries
  - $P(D|I) = \sum_i p(D|q_i)p(q_i|I)$

Transformed Queries and Search

- Generate a “better” query
- Generate queries and test which appears to work best
- Generate a ranking of queries and combine results
- Use transformation model to develop a relevance or information need model, incorporate into retrieval model

Transforming Long Queries

- Long queries occur in many applications
  - e.g., CQA, forums, professional, even Web
- Long queries may be the best way of expressing most information needs
  - i.e., selecting keywords can be difficult for people
- Long queries represent the next stage in evolution of search engines

Long Query Examples

- TREC description query
  - e.g., *"Provide information on all kinds of material international support provided to either side in the Spanish Civil War."*
- Questions from users in CQA services
  - e.g., *"Where can I complain about my wedding photographer?"*

Long Query Examples

- Queries with more than one keyword or phrase from web logs
  - e.g., *"lessons about kids in the bible", "best time of the year to visit bolivia"*
- Whole sentences or passages from documents
  - e.g., *"Process for the preparation of a zeolitic catalyst which comprises treating a zeolite of the Y-type having an alkali metal oxide/aluminium oxide molar ratio of at most 0.13 with a solution of a multi-valent metal salt having a cationic radius between 0.6 and 1.0 angstrom and combining the ion-exchanged zeolite without a calcination step with a hydrogenation component of a Group 8 and/or Group 6b metal."*

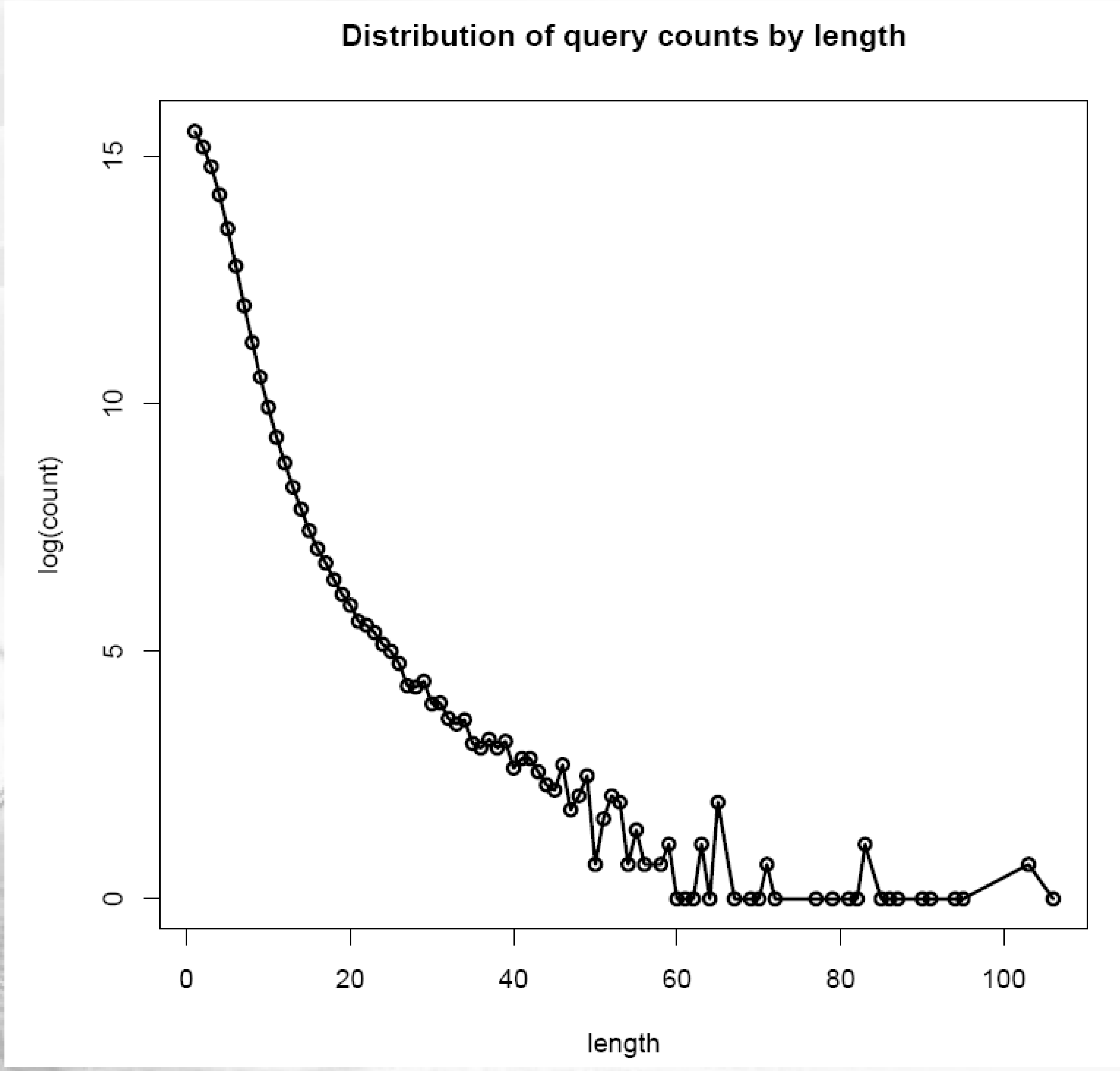
Characteristics of a Long Query

- Length (duh!)
  - Average length of Q&A questions more than 20 words and about 9 words for FAQs from Web
  - TREC descriptions are 14-20 words average vs. 2.5-5 words for title
- Grammar
  - Long queries tend to be more grammatical, sometimes full sentences
  - But, from a Q&A log:
    - "who the first one fly to the spase"

Characteristics of a Long Query

- Frequency
  - Duplicates of long queries are generally rare
  - So, long queries are part of the "long tail"
- Information need
  - More complex information needs?
    - or maybe a better expression of real information needs than keywords
  - Usually not homepage/navigational searches
    - but sometimes used for known-item searching

MSN Query Log



- Queries of length 4 or less account for 90.3%
- Average query length is 2.4

MSN Query Log

- Long query types
  - *Questions* (e.g., wh-)
  - *Operators* (contains query language operators)
  - *Composite* (made up of short queries)
  - *Non-Composite* (noun phrases and sentences)
  - *Exact quotes*

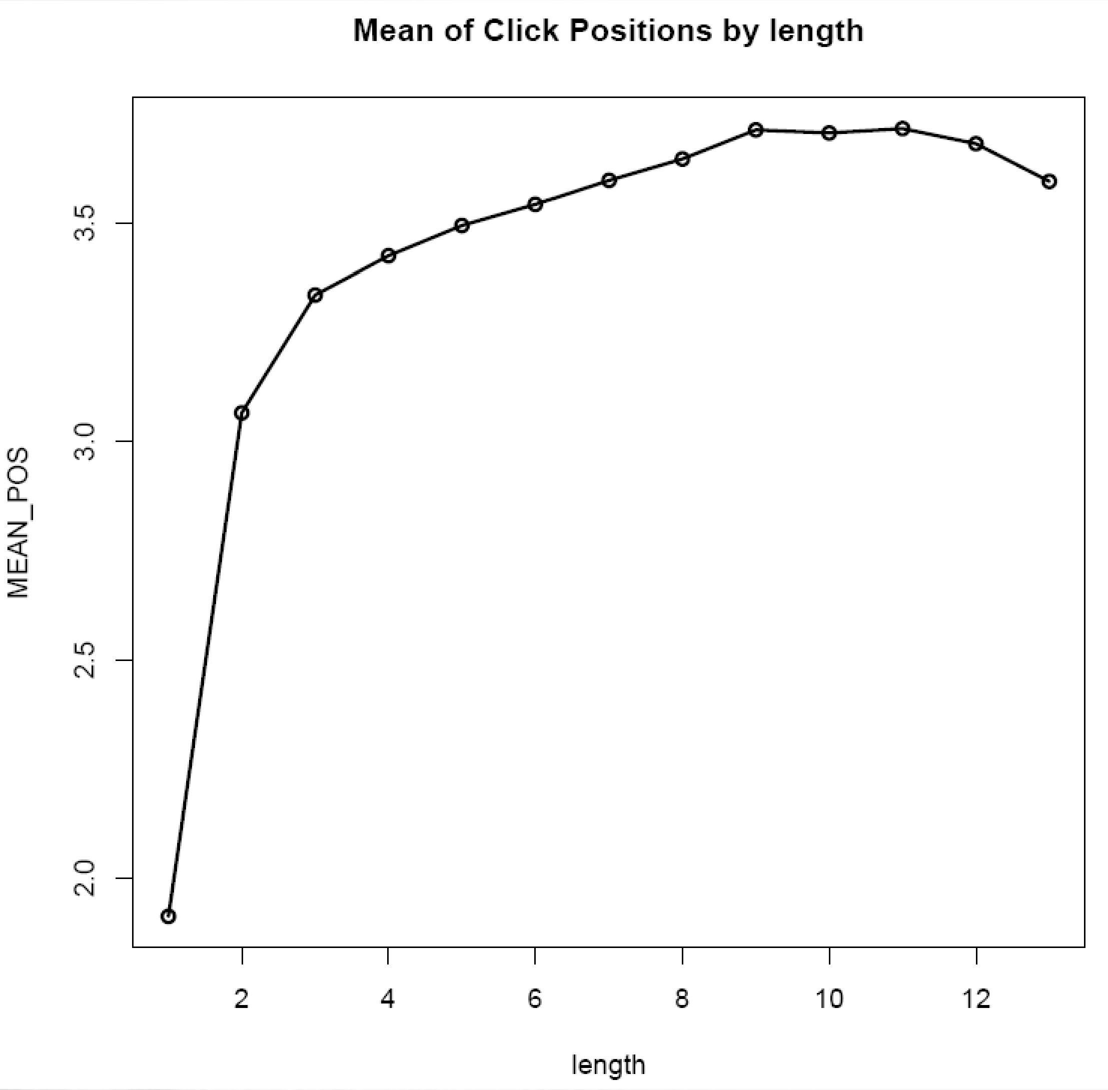
MSN Query Log

Total Queries: 14,921,286		
Long Queries ($5 \leq l(q) \leq 12$) **:** 1,423,664		
Type	Count	% of Long
Questions	106,587	7.49
Operators	78,331	5.50
Composite	918,482	64.52
Non-Composite	320,263	22.50
Noun-Phrases	*204,823*	*14.39*
Pseudo-Sentences	*115,440*	*8.11*

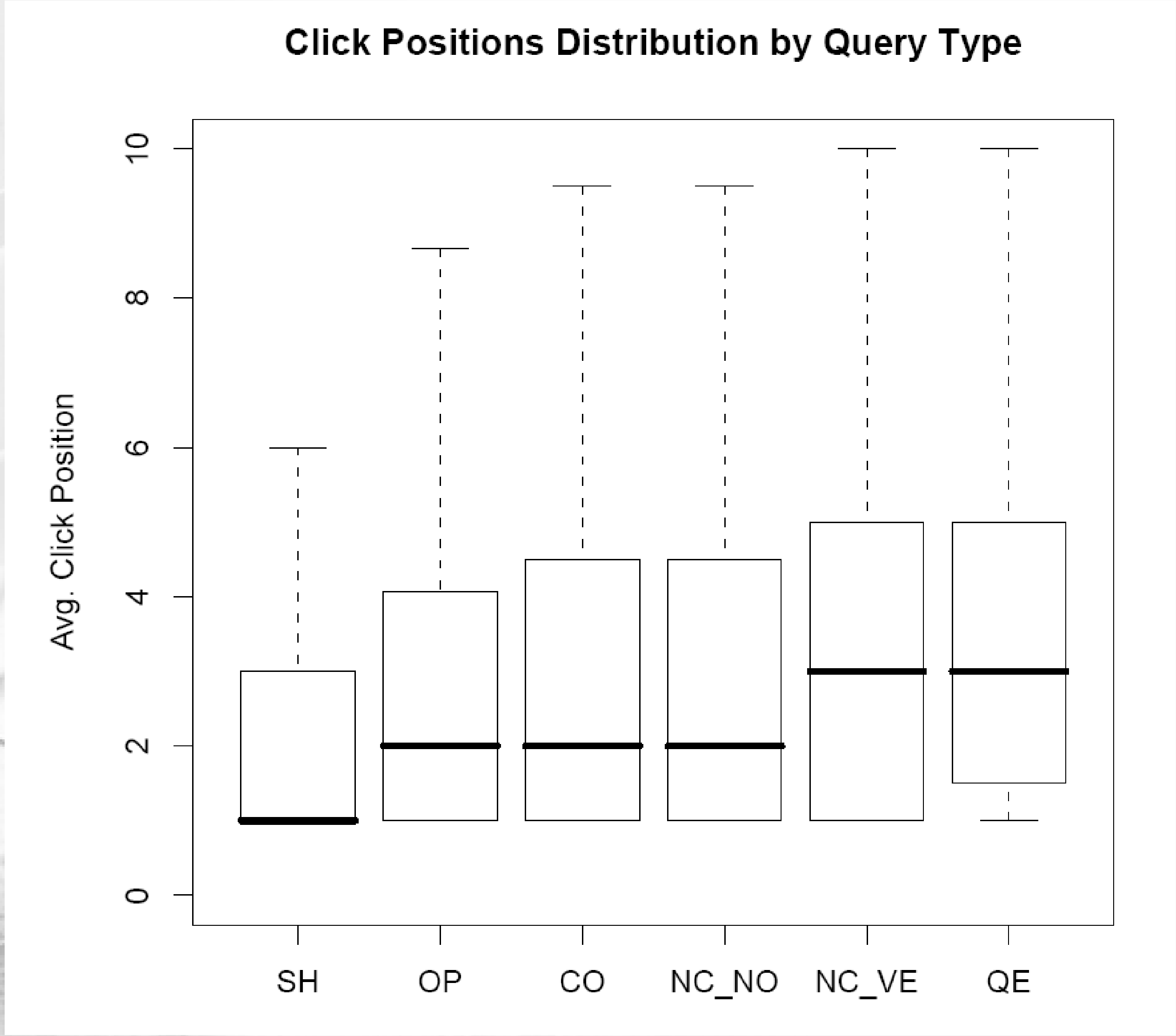
Do Long Queries Work?

- For people, yes; for search engines, no
- Long queries give generally poor, unpredictable results with current Web search engines
- TREC description queries don't work as well as title queries
- QA techniques don't work well for more general questions

MSN Query Log



MSN Query Log



Long Query Transformations

- Adding structure
  - e.g., query segmentation, identify key concepts, linguistic features, document structure features, ignore or reduce weight of some parts
- Finding or generating similar queries
  - e.g., translation models, paraphrasing, expansion

Finding key concepts

- Long or “verbose” queries mix key concepts with additional qualifications, relationships, structure
- Current search engines don’t make good use of this additional text
- Goals
  - Develop techniques to identify key concepts in queries
  - Transform queries using concepts

Concepts?

- Everything is potentially a concept
  - single words, nouns, phrases, named entities, verbs, etc.
- Simple noun phrases used in Bendersky and Croft (2008)
  - nouns generally most important part of queries
  - common assumption in previous work (although some verbs are important)

TREC example

Provide information on all kinds of material international support provided to either side in the Spanish Civil War

<u>Concept extraction</u>

[information, kinds, material international support, side, Spanish Civil War]

Concept Weighting

- Two approaches:
  - Unsupervised – estimate importance using concept IDF
  - Supervised: Train a classifier to recognize key concepts, weight by estimate of probability that concept belongs to that class

Collection-based features

is_cap(c_i) - Is concept capitalized?

tf(c_i) - Concept TF in the collection

idf(c_i) - Concept IDF in the collection

ridf(c_i) - Concept residual IDF in the collection (*Actual IDF deviation from Poisson model prediction; Church & Gale, 1995*)

wig(c_i) - Concept Weighted Information Gain *(Zhou & Croft, 2007)*

Collection-independent features

g_tf(c_i) - Concept frequency in *Google n-grams*. Estimates concept frequency in a large web collection

qp(c_i) - Number of times a concept was used as a part of a query, extracted from *Live Search* query logs

qe(c_i) - Number of times a concept was used as an exact query, extracted from *Live Search* query logs

Concept classification

- Train a classifier on a set of labeled concept instances
- Training data generated by annotation, not by using title queries
  - assumed only one key concept per query
  - title queries can contain words not in description

Transformed queries

- Baseline
  - title:
    - #combine(Spanish Civil War support)
  - description:
    - #combine(information kinds material international support provided side Spanish Civil War)
- Key concept
  - weighted combination:
    - #weight(
      0.8 #combine(information kinds material international support
      provided side Spanish Civil War)
      0.2 #weight(0.99994 #combine (Spanish Civil War) 0.00006
      #combine (material international support)))

Transformed Queries

- Dependence model
  - #weight(
    0.85 #combine(information kinds material international support
    provided side Spanish Civil War)
    0.10 #combine(#od:1(information kinds) #od:1(kinds material)
    #od:1(material international) #od:1(international support)
    #od:1(support provided) #od:1(provided side)
    #od:1(side Spanish) #od:1(Spanish Civil)
    #od:1(Civil War))
    0.05 #combine(#uw:8(information kinds) #uw:8(kinds material)
    #uw:8(material international) #uw:8(international support)
    #uw:8(support provided) #uw:8(provided side) #uw:8(side
    Spanish) #uw:8(Spanish Civil) #uw:8(Civil War)))

Retrieval results

	ROBUST04		W10g		GOV2	
	prec@5	MAP	prec@5	MAP	prec@5	MAP
<title>	47.80	25.28	30.73_d	19.31	56.75	$\mathbf{29.67}_d$
<desc>	47.26	24.50	39.20^t	18.62	52.62	25.27^t
SeqDep<desc>	**49.11**	25.69_d	39.80^t	19.28	$\mathbf{56.88}_d$	27.53^t_d
KeyConcept[2]<desc>	48.54	$\mathbf{26.20}_d$	$\mathbf{40.40}^t$	$\mathbf{20.46}^t_d$	56.77_d	27.27^t_d

MAP and Precision at 5 results.

Next Steps

- Revisit concepts
  - use relevance data to train weights for *all* query terms
  - Regression Rank (Lease et al, this conference)
- Integrate with stemming, segmentation
  - e.g. “material international support”
- Infer relationships with document structure
  - e.g., Kim et al, this conference
  - e.g., Petkova and Croft, this conference

Structure Inference Example

#	content-only	content-and-structure queries	scores
Q5	*receptionist microsoft office arizona*	//resume[.//desiredjobtitle[~'receptionist']][.//skillname[~'microsoft office']] [.//state[~'arizona']]	1.000000
		//resume[.//desiredjobtitle[~'office receptionist']][.//skillname[~'microsoft']] [.//state[~'arizona']]	0.691925
		//resume[.//title[~'receptionist']][.//skillname[~'microsoft office']] [.//state[~'arizona']]	0.633407
Q6	*emergency room registered nurse mesa az with license*	//resume[.//resumetitle[~'registered nurse']][.//title[~'emergency room']] [.//educationsummary[~'license']] [.//city[~'mesa']][.//stateabbrev[~'az']]	1.000000
		//resume[.//title[~'emergency room']][.//resumetitle[~'registered nurse']] [.//additionalinfo[~'license']] [.//city[~'mesa']][.//stateabbrev[~'az']]	0.939304

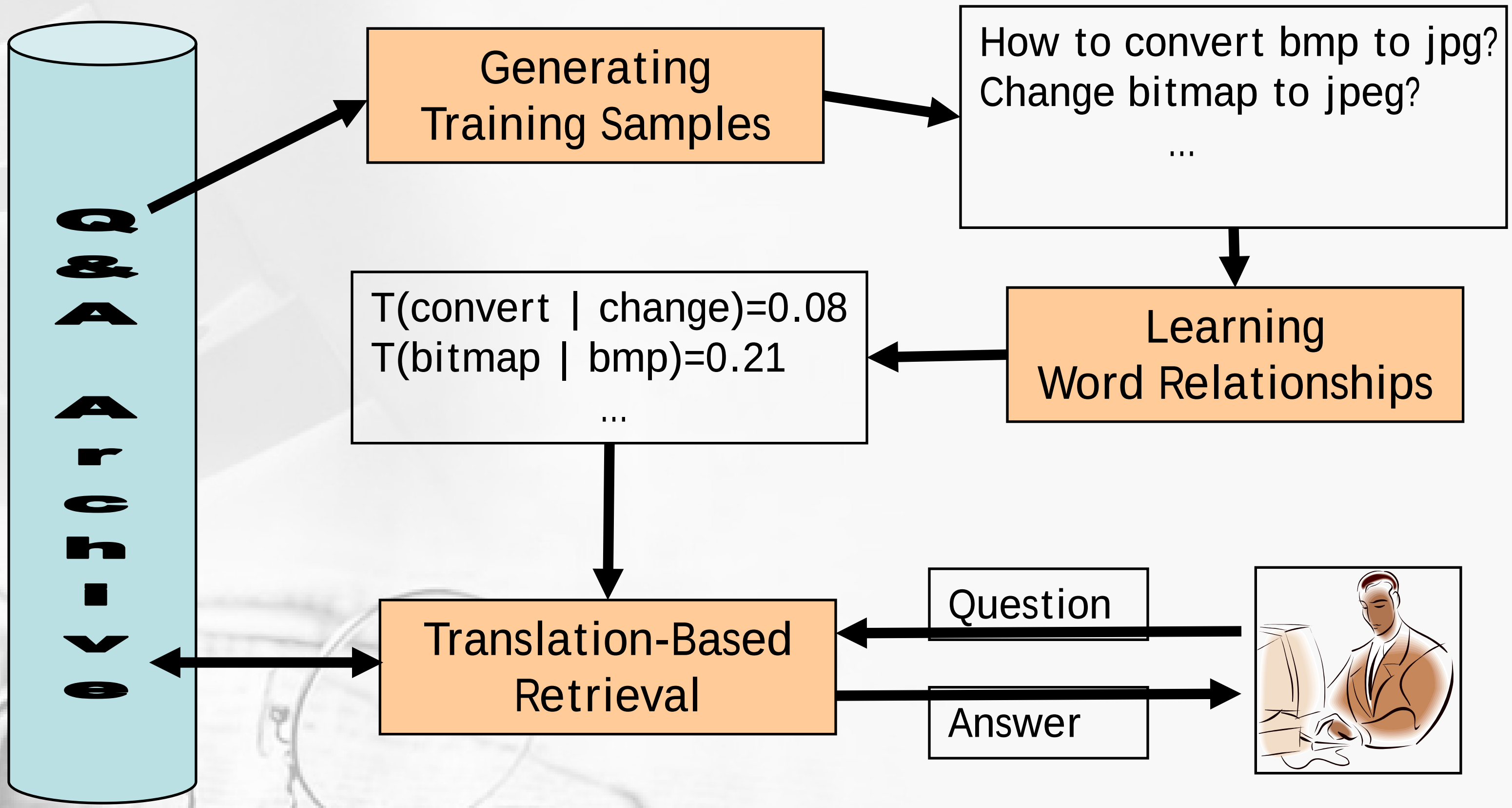
Translation and Transformation

- Statistical translation models are basis of machine translation
- Berger and Lafferty used “noisy channel” model to describe language model approach to retrieval
  - simple translation model from document to query
  - main problem is estimation of translation probabilities (no parallel corpora)

Translation and Transformation

- Translation model can also be used to find (or generate) similar queries
  - simple model describes word or phrase substitution
  - could also model ordering, fertility, other structure
- Finding similar long queries is a key problem in collaborative question answering (CQA)

Overview of CQA search



Estimating Translation Probabilities

- First approach: Generate pairs of semantically related questions using answer similarity
  - similar to query clustering using clickthrough
- Second approach: Use question and answer pairs directly
  - uses EM-based algorithm from IBM model 1
  - in case of Q&A pairs, either question or answer can be used as source or target

Examples of generated query pairs

Can I attach a 5 mega byte file in my email? Sending big movie files to my friends over the net by email.
Why do we have to use only English for email addresses? Why can't I use Korean in email IDs?
What is the best email service? Who provides the most popular and powerful email accounts?
Who invented email? The first person who used email?

Naver data, 5,200 Q&As from "email" category

Translation examples

P(A|Q)	P(Q|A)	P_{pool}
everest	mountain	everest
29,035	tallest	mountain
ft	everest	tallest
mount	highest	29,035
8,850	mt	highest
feet	discover	mt
measure	hillary	ft
expedition	edmund	measure
height	mountin	feet
nepal	biggest	mount

Top 10 translations for "everest" estimated from Wondir data

Question retrieval results

Model	Trans. Prob.	MAP	P@10
LM	-	0.322	0.221
RM	-	0.340	0.240
TransLM	P(A|Q)	0.406	0.268
TransLM	P(Q|A)	0.379	0.266
TransLM	P_{pool}	0.424	0.287

Wondir data, 50 TREC QA queries

Examples of Question Retrieval

TransLM+QL
Who is the leader of India?
who is the prime minister of india
who is the current vice prime minister of india
who is the army chief of india
who is the first prime minister of india
Who made the first airplane that could fly?
what is the oldest airline that still fly airplane
who was the first one who fly with plane
who was the first person to fly a plane
who the first one fly to the spase
who the first one to fly to sky

“ASK” examples

Query	*What did Vasco da Gama discover?*	LM Rank	Trans Rank
Question	why was portugal able to take an early lead in the exploration of the indian ocean	X	12

Query	*What was the name of the famous battle in 1836 between Texas and Mexico?*	LM Rank	Trans Rank
Question	how did the battle od the alamo start	230	17

Query	*What is a caldera?*	LM Rank	Trans Rank
Question	what is the open at the top of a volcano call	X	15

Query	*Where is the Danube?*	LM Rank	Trans Rank
Question	what river flow from germany to hungary to the black sea the answer start with the letter d	X	15

Next Steps

- Explore the translation model for query generation and modeling in more general retrieval situations
- Test new estimation techniques for translation probabilities
  - e.g., n-grams, similar sentences, anchor text, web query logs
- Integrate with other transformation processes

Translation from N-grams

- Treat n-grams with similar content as translation pairs

$$P(w_1, w_2) = \sum_c P(w_1, w_2, c) = \sum_c P(w_1|c)P(w_2|c)P(c)$$

  - w_1, w_2 are words or phrases, c is context
  - note that this is same as relevance model but with different context
  - for n-gram $(w_1, w_2, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_{n-1}, w_n)$, context for w_i is $(w_1, w_2, \ldots, w_{i-1}, \text{SPACE}, w_{i+1}, \ldots, w_{n-1}, w_n)$

Examples

```
companies:226
firms:0.107626 corporations:0.024174 sides:0.0164981 makers:0.0132416 firm:0.012543
resume:0.010891 businesses:0.0104147

aircraft:224
jets:0.0350821 airplane:0.0300076 airplanes:0.0262349 airliner:0.0249716
planes:0.0236632 plane:0.0188136 helicopter:0.0160031 craft:0.00904625

conflict:227
conflicts:0.0409008 clash:0.0134826 crisis:0.011075 dispute:0.00911446
strife:0.00833837 topic:0.00758214 differences:0.00695771 tensions:0.00679373

conflicting:225
differing:0.0307591 contradictory:0.0257209 different:0.0211121
persistent:0.0190258 varying:0.0172985 unconfirmed:0.0162109 mixed:0.0149626

constitution:255
clause:0.01437 statute:0.00900882 charter:0.00871149 amendment:0.00823871
reorganization:0.00819322 contract:0.00769068 consitution:0.00762612
code:0.00738253 provision:0.00729247

contracts:253
contract:0.0350073 pacts:0.0262955 agreements:0.0231466 futures:0.0205652
soybeans:0.0127076 products:0.0102952 exchanges:0.00974947 approval:0.00876252
```

Translation from N-grams

- More like synonyms than expansion from relevance model
  - also narrower than terms from CQA experiments
- Transformation needs to incorporate both substitution and expansion
  - cf. Wang and Zhai, CIKM 2008

Summary

- Query transformation can unify many processes that have been addressed separately
  - goal is to build a better relevance model, not just suggest queries
- Long queries offer more challenges and rewards
- Much more work needs to be done on linguistic features, relationships, etc.
  - may need framework with more ability to do inference

Advertisement

Lemur Query Log Project

- Goal: Collect a query log using Lemur toolbar that can be shared with academic researchers

http://lemurstudy.cs.umass.edu/