IN4325
Query refinement

Claudia Hauff (WIS, TU Delft)

The big picture

The essence of IR

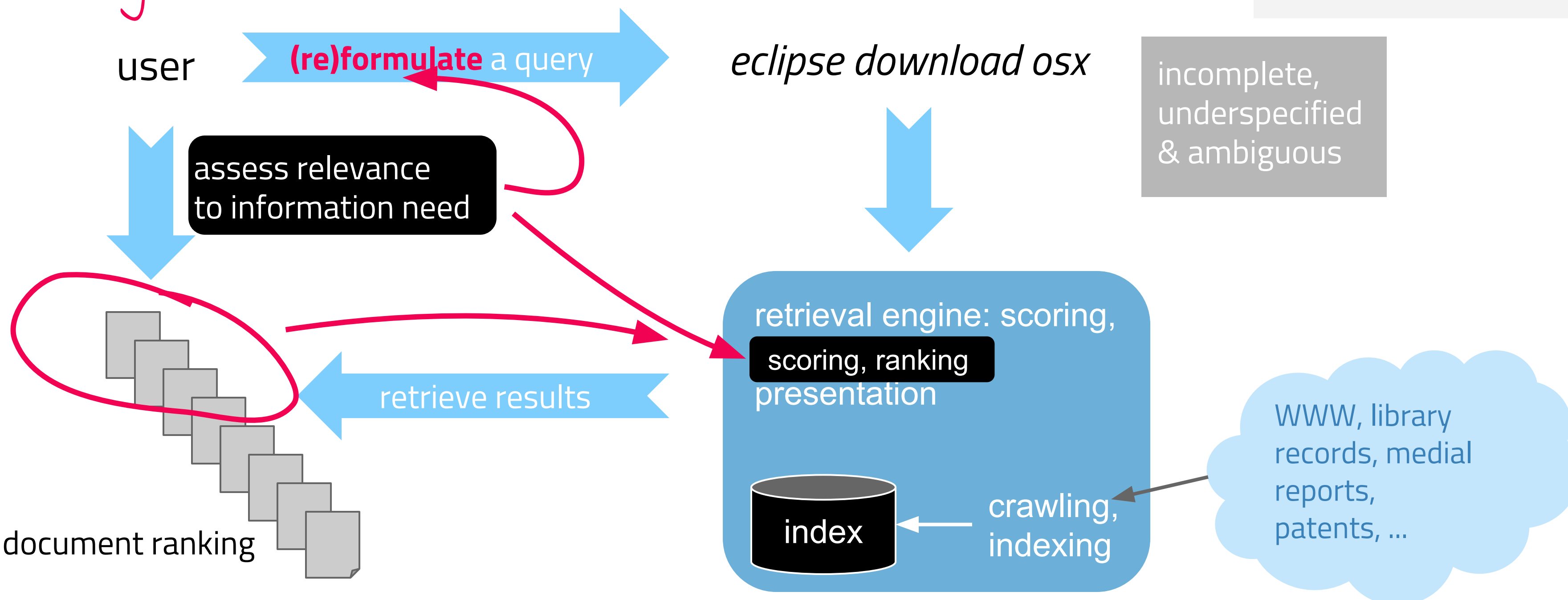
Information need
Topic the user wants to know more about

Query
Translation of need into an input for the search engine

Relevance
A document is relevant if it (partially) provides answers to the information need

Information need: *Looks like I need Eclipse for this job. Where can I download the latest beta version for macOS Sierra?*

today's focus



Information needs

Different categorizations exist:

- **Informational** vs. **transactional** vs. **navigational**
- Number of relevant documents wanted
- Tasks underlying the information need

Belkin's **Anomalous State of Knowledge**: users do not always know what exactly their information need is

today's web search queries are 2-3 terms long

Thus:

- Queries can represent different information needs (short, ambiguous, imprecise)
- A query may be a poor representation of the underlying information need

Query refinement techniques

Query refinement either automatically or through user interactions

- Query expansion
- (Pseudo-) relevance feedback

- Spelling correction
- Query autocompletion
- Query suggestions

Goal: produce a query that is a **better representation** of the information need (this in turn *should* lead to a better set of retrieved documents)

Query expansion


Semantic gap
flickr@nolatularosa

Query expansion

Idea: instead of a user manually adding synonyms to her query, let the system help (automatically or semi-automatically) in order to decrease the **semantic gap**.

Global approaches (independent of the query)

- Query expansion with a domain-specific thesaurus (indexing vocabulary and simple relations) is common and successful for domain-specific corpora
- A generic “thesaurus” such as WordNet has not shown to be effective

Local approaches (relative to the retrieved documents)

- Relevance feedback
- Pseudo-relevance feedback
- Implicit feedback

tank

All | Images | Videos | Maps | News | My saves

18.900.000 Results | Date | Only English | Region

Videos of tank

bing.com/videos

Tank - Next Breath [Music Video]
YouTube · 28-2-2012 · 7M+ views

Tank - Better For You [Official Video]
YouTube · 23-12-2015 · 2M+

Tank - I Can't Make You L Me [Official Music Video]
YouTube · 15-12-2010 · 51M

See more videos of tank

World of Tanks | Epic Online Tank Game | Play for Free

https://worldoftanks.com

Furious 15-vs-15 Battles on Legendary Tanks, Over 500 War Vehicles are Ready to Roll Out. Join Multiplayer **Tank** Game with 150 Million Players Worldwide!

Tank (1984) - IMDb

www.imdb.com/title/tt0088224

Jaegers, assassins, and superheroes await you in our Winter Movie Guide. Plan your season and note of the hotly anticipated indie, foreign, and documentary ...

5,6/10 ★★★☆☆ (3,4K) **Category:** Comedy

Cast: James Garner/Shirley Jones/C. Thomas… **Content Rating:** PG

TANK ARCHITECTURE AND INTERIOR DESIGN

tank.nl

TANK is an international design studio for architecture, interior design and branding. **TANK** realises bespoke and outstanding projects.

Images of tank

bing.com/images

See more images of tank

Alzheimer Disease MeSH Descriptor Data 2018

Details | Qualifiers | MeSH Tree Structures | Concepts

MeSH Heading	Alzheimer Disease
Tree Number(s)	C10.228.140.380.100 C10.574.945.249 F03.615.400.100
Unique ID	D000544
Scope Note	A degenerative disease of the BRAIN characterized by the insidious onset of DEMENTIA. Impairment of MEMORY, judgment, attention span, and problem solving skills are followed by severe APRAXIAS and a global loss of cognitive abilities. The condition primarily occurs after age 60, and is marked pathologically by severe cortical atrophy and the triad of SENILE PLAQUES; NEUROFIBRILLARY TANGLES; and NEUROPIL THREADS. (From Adams et al., Principles of Neurology, 6th ed, pp1049-57)
Entry Version	ALZHEIMER DIS
Entry Term(s)	Acute Confusional Senile Dementia Alzheimer Dementia Alzheimer Disease, Early Onset Alzheimer Disease, Late Onset Alzheimer Sclerosis Alzheimer Syndrome Alzheimer Type Senile Dementia Alzheimer's Disease Alzheimer's Disease, Focal Onset Alzheimer-Type Dementia (ATD) Dementia, Alzheimer Type

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: tank [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **tank**, army tank, armored combat vehicle, armoured combat vehicle (an enclosed armored military vehicle; has a cannon and moves on caterpillar treads)
- S: (n) **tank**, storage tank (a large (usually metallic) vessel for holding gases or liquids)
- S: (n) **tank**, tankful (as much as a tank will hold)
- S: (n) tank car, **tank** (a freight car that transports liquids or gases in bulk)
- S: (n) cooler, **tank** (a cell for violent prisoners)

WordNet has been tried and tested

Are the glosses useful for anything?

Noun

- S: (n) **Pluto** (a cartoon character created by Walt Disney)
- S: (n) **Pluto**, Dis, Dis Pater, Orcus ((Roman mythology) god of the underworld; counterpart of Greek Hades)
- S: (n) **Pluto** (a large asteroid that was once thought to be the farthest known planet from the sun; it has an elliptical orbit) *"Pluto was discovered by Clyde Tombaugh in 1930"*

Ambiguous query: "Pluto"

20+% of Web search queries are single term queries. Very common in site search too!

Idea: give the user a choice between possible hypernyms if we do not have a query log

Bootstrapping how?

1. Look up WordNet synsets and glosses
2. Process the glosses (POS tagger, etc.) and keep the nouns as potential hypernyms
3. Apply **Hearst patterns** and retrieve the number of result pages per candidate
4. Consider the candidate with the highest score as hypernym

simple patterns to decide on taxonomic relations

- NP_0 *such as* {NP_1,NP_2,…, (*and*|*or*)} NP_n
  - "American cars such as " ➔ Chevrolet, Pontiac
- *Such NP as* {*NP*, }* {(*or*|*and*)} *NP*
  - "such colors as red or orange"
- *NP* {, *NP*}* {,} *or other NP*
- *NP* {, *NP*}* {,} *and other NP*
- *NP* {,} *including* {*NP*,}* {*or*|*and*} *NP*
- *NP* {,} *especially* {*NP*,}* {*or*|*and*} *NP*

WordNet has been tried and tested

1. WordNet glosses for query term "Pluto"

- **SYN1** a small planet and the farthes known planet from the sun; has the most elliptical orbit of all the planets
- **SYN2** (Greek mythology) the god of the underworld in ancient mythology; brother of Zeus and husband of Persephone
- **SYN3** a cartoon character created by Walt Disney

2. Candidate nouns

- **SYN1** planet, sun, orbit, planets
- **SYN2** Greek, god, underworld, mythology, brother, Zeus, husband, Persephone
- **SYN3** cartoon, character, Walt, Disney

3. Hearst patterns and page counts (shown for SYN1 only)

- "Pluto is a planet" (1550), "Pluto is planet" (145) "Pluto is a sun" (2), "Pluto is sun" (0)
- "Pluto is an orbit" (1), "Pluto is orbit" (1) "Pluto is a planets" (0), "Pluto is planets" (0)

4. Refinement offers: "Pluto planet", "Pluto god", "Pluto cartoon"

http://ceur-ws.org/Vol-176/paper13.pdf

Pseudo-relevance feedback

PRF: we assume the top-ranked documents are relevant
RF: user indicates which top-ranked documents are relevant

information need

topic

query

search session

Search session
A set of searches conducted by a user to solve a (complex) search task within a limited amount of time.

Overview

Approach:

- User issues a short query
- System returns an initial set/ranked list of results
- **User marks some of the results relevant or non-relevant**
- System computes a **better representation** of the information need based on the feedback
- System displays revised set/ranked list of results

loop

training data of a search session

machine learning with very limited data

Insight: it is difficult to formulate a good query based on a complex information need, but it is relatively easy to **decide** whether the returned documents match the information need

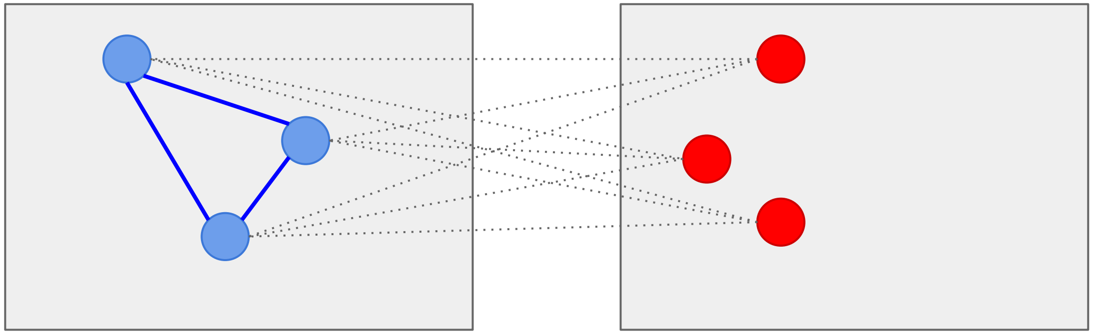
(P)RF implementation is **retrieval model dependent**;
Strategy: words that occur more frequently in relevant than non-relevant documents are added to the query or increased in weight.

Relevance feedback builds on the cluster hypothesis

"*Closely associated documents tend to be relevant to the same requests.*" (Keith van Rijsbergen, 1970s)

Common assumption of IR systems: relevant documents are more similar to each other than to non-relevant documents

association between all document pairs (R-R), (R-NR)

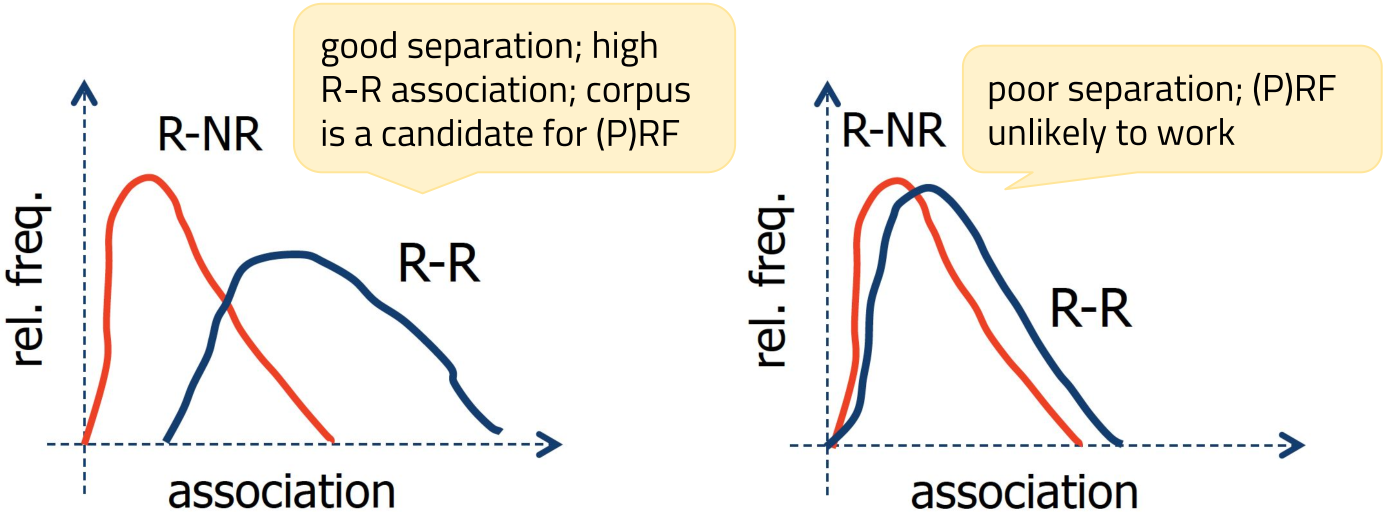


relevant documents (R) non-relevant documents (NR)

Relevance feedback builds on the cluster hypothesis

"*Closely associated documents tend to be relevant to the same requests.*" (Keith van Rijsbergen, 1970s)

Plot the relative frequency (binned) against the strength of association (usually cosine similarity)



Relevance feedback builds on the cluster hypothesis

Clustering methods should:

- Produce a **stable clustering**: no sudden changes when items are added/removed
- Be **tolerant to errors**: small errors should lead to small changes in clustering

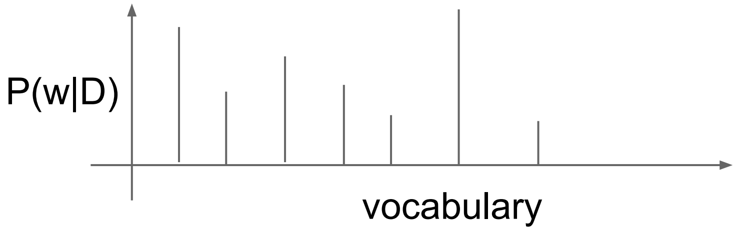
Clustering fails when:

- Subset of documents have very different important terms (semantics to the rescue!)
- Queries are inherently disjunctive
- Polysemy occurs

Pseudo-relevance feedback in language models

Language models

- **Unigram language model**: probability distribution over the words (the *vocabulary*) in a language (the *collection* or *document*)
- In IR, unigram LMs represent the *topical content*



- A LM representation of a document can be used to *generate* new text by sampling terms from the distribution (the text won't have a syntactic structure, but that's fine)

Language models

Smoothing

General idea: discount probabilities of **seen words**, assign extra probability mass to **unseen words** with a fallback model (the *collection language model*)

$$P(w \mid D) = \begin{cases} P_{smoothed}(w \mid D) & \textit{if word } w \textit{ is seen} \\ \alpha_d P(w \mid \mathbb{C}) & \textit{otherwise} \end{cases}$$

Jelineck-Mercer (JM) smoothing: linear interpolation (amount of smoothing controlled) between ML and collection LM

$$P_\lambda(w \mid D) = (1-\lambda)P_{ml}(w \mid D) + \lambda P(w \mid \mathbb{C}), \quad \lambda \in (0,1)$$

Language models

Smoothing

General idea: discount probabilities of **seen words**, assign extra probability mass to **unseen words** with a fallback model (the *collection language model*)

$$P(w \mid D) = \begin{cases} P_{smoothed}(w \mid D) & if\ word\ w\ is\ seen \\ \alpha_d P(w \mid \mathbb{C}) & otherwise \end{cases}$$

Dirichlet smoothing: longer documents receive less smoothing

"count" of term w in D

$$P_\mu(w \mid D) = \frac{c(w;D) + \mu P(w \mid \mathbb{C})}{\sum_w c(w;D) + \mu},\ usually\ \mu > 100$$

Model generalization: create a model/framework that contains existing models as special cases

flickr@marcyleigh

Relevance models

- Query is a **fixed sample** in LM, with documents being ranked according to their prob. of generating the sample

- Relevance feedback does not come naturally to LM
  - BIM: adjust the weights of the relevance set
  - VSM: Rocchio

- Idea: instead of a fixed sample, consider the query to be a language model (the **relevance model**); it represents the topic covered by relevant documents
  - RF is now principled!

query text now a very small sample generated from the relevance model;
relevant documents are larger samples from the same model

https://dl.acm.org/citation.cfm?id=383972

Relevance models

Two options to use our relevance model R

- Option 1: Rank documents by $P(D|R)$
- Option 2: Rank documents according to their *similarity* between the document LM and the query (relevance) LM

Difficult for diverse (wrt. length, vocabulary) sets of documents.

not symmetric

Kullback-Leibler divergence ("KL divergence") measures the difference between two probability distributions P and Q:

$$KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

"true distribution"; usually R

always positive (larger the more apart two distributions are); we use the **negative KL divergence** to rank

https://dl.acm.org/citation.cfm?id=383972

Relevance models

$$\log\frac{M}{N} = \log M - \log N$$

relevance model

$$-KL(P||Q) = -\sum_x P(x)\log\frac{P(x)}{Q(x)}$$

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

independent of the document, ignore for ranking purposes

Maximum likelihood estimate of $P(w|R)$

$$= \sum_{w\in V}\frac{f_{w,Q}}{|Q|}\log P(w|D)$$

We can ignore terms not in Q.

Isn't this rank equivalent to query likelihood? Yes!

However: we have a **more general model**, we can estimate the relevance model in many ways!

https://dl.acm.org/citation.cfm?id=383972

Relevance models

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

$$P(w|R) \approx P(w|q_1, q_2, .., q_n)$$

$$P(w|R) \approx \frac{P(w, q_1, q_2, .., q_n)}{P(q_1, q_2, .., q_n)}$$

if query terms are samples from the relevance model, an unseen word's probability should depend on the query terms

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

set of language models

term independence assumption

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

https://dl.acm.org/citation.cfm?id=383972

Relevance models

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

$$P(w|R) \approx P(w|q_1, q_2, .., q_n)$$

$$P(w|R) \approx \frac{P(w, q_1, q_2, .., q_n)}{P(q_1, q_2, .., q_n)}$$

if query terms are samples from the relevance model, an unseen word's probability should depend on the query terms

$$P(w, q_1, q_2, .., q_n) = \sum_{D \in \mathcal{C}} \underbrace{P(D)}_{\text{prior probability of a document}} P(w|D) \underbrace{\prod_{i=1}^{n} P(q_i|D)}_{\text{query likelihood score of a document}}$$

Requires **two passes for ranking**:

1. Rank documents using query likelihood to obtain the weights needed
2. Use KL-divergence to rank documents by comparing the relevance model and document model

i.e. pseudo-relevance feedback (formally in LM)

Relevance models

In the literature often referred to as RM1, RM2, **RM3**, RM4.

Query drift
The presence of aspects/topics not related to the query in the top-retrieved documents.

Once more ...

1. Rank documents using the query likelihood score for query Q.
2. Select some number of the top-ranked documents to be the set C.
3. Calculate the relevance model probabilities $P(w|R)$ using the estimate for $P(w, q_1 \ldots q_n)$.
4. Rank documents again using the KL-divergence score:[13]

The whole collection? Just the top 10-50 ranked ones?

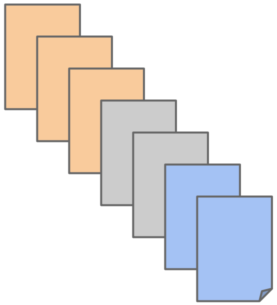
$$\sum_w P(w|R) \log P(w|D)$$

All vocabulary terms? Just the 10-25 that have the highest probabilities?

Actually, this model is well motivated but in practice a slight adaptation has turned out to give the best results (=RM3):

Interpolate the relevance model with the original query model to avoid **query drift**.

Relevance models and clustering



Nothing stops us from smoothing the document language model with **document clusters**:

$$P(w|D) = \lambda P_{ML}(w|D) + (1-\lambda)P(w|Cluster)$$
$$= \lambda P_{ML}(w|D) + (1-\lambda)[\beta P_{ML}(w|Cluster) + (1-\beta)P_{ML}(w|Coll)]$$

Many decisions: which clustering algorithm? How many clusters? Clustering (in)dependent of the queries?

http://ciir.cs.umass.edu/pubfiles/ir-347.pdf

Negative relevance feedback

Another common way of denoting a language model of the **Q**uery and **D**ocument

$$S(Q,D) = -D(\theta_Q||\theta_D) = -\sum_{w \in V} p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|\theta_D)}$$

KL-divergence

Negative feedback is easy to integrate into the vector space model (remember Rocchio).

In language modeling, it is less natural to directly modify the relevance model (neg. probabilities are not possible).

Idea: $S(Q,D) = -D(\theta_Q||\theta_D) + \beta \cdot D(\theta_N||\theta_D)$

Create a single negative topic model and penalize the document score if it is similar to it..

https://dl.acm.org/citation.cfm?id=1390374

Estimating relevance models based on external corpora

Idea: mixture of relevance models drawn from different corpora:

$$P(w|\hat{\theta}_Q) = \sum_{c \in \mathcal{C}} P(c) P(w|\theta_Q, c)$$

External corpus only

Mixture of external corpus and target corpus

			BIGNEWS		GOV2		WEB	
	QL	RM3	EE	MoRM	EE	MoRM	EE	MoRM
trec12	0.2502	0.3201	0.3204	0.3319	0.2709	0.3215	0.3092	0.3324
robust	0.2649	0.3214	0.3501	0.3530	0.2748	0.3207	0.3301	0.3352
wt10g	0.1982	0.2030	0.2256	0.2331	0.1999	0.1958	0.2452	0.2429

Mean average precision

collection	docs	terms
BIGNEWS	6,422,629	2,417,464
GOV2	25,205,179	49,917,419
WEB	19,200,000,000	-

https://dl.acm.org/citation.cfm?id=1148200

Last words on query expansion

Has not been taken up by Web search engines

- WSEs cannot afford **computationally expensive** AQE techniques (millisecond response times required)

- AQE techniques **perform well on average**, but can cause severe degradation for some queries
- AQE tends to improve **recall** (instead of guaranteeing high precision), often less important for WSE

- **Users** may get confused (their query does not match the returned results)

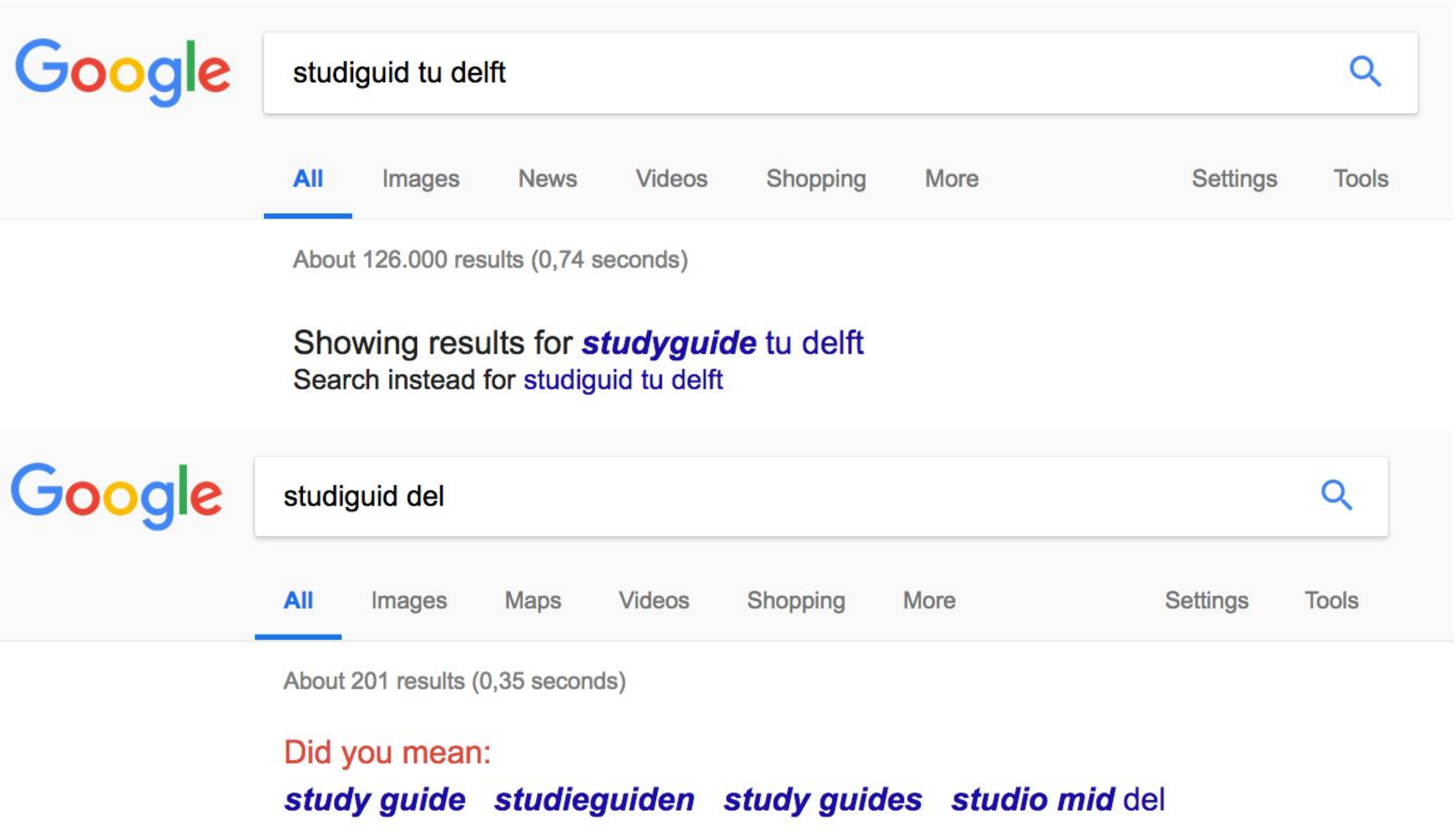
Last words on query expansion

Common applications of automatic query expansion (besides document ranking):

- **Question answering**: retrieving passages of documents containing answers to concrete questions, e.g. “When was Barack Obama born”?
- **Multimedia IR**: text-based search over media metadata (annotations, concepts, speech transcripts) as well as multimodal search
- **Information filtering**: monitoring a stream of documents and selecting those that are relevant to a user
- **Cross-language IR**: retrieving documents written in other languages than the query’s language

Spell checking

Web search: “Did you mean …”



Overview

`extenssions` → `extensions` (insertion error)
`poiner` → `pointer` (deletion error)
`marshmellow` → `marshmallow` (substitution error)
`brimingham` → `birmingham` (transposition error)
`doceration` → `decoration` (2 substitution errors)

10-15% of Web search queries contain spelling errors; most are single-character errors

Challenges: variety in type and severity of possible spelling errors in queries (little context available); no definite lexicon (**Heap's law**)

Generic spell checker:

- Create a spelling dictionary and suggest corrections for any word w not in it
- Suggestions based on similarity between dictionary words and w
  - Levenshtein edit distance
  - Soundex

Assumptions in practice:
- first letter is correct
- correct term has similar length

A hundred years old ...

Soundex

Homophone: word that is pronounced the same way as another word but differs in meaning (e.g. *raise* vs. *rays*)

Soundex is a **phonetic encoding** originally employed for name matching

```
extenssions → E235
extensions → E235
```

Use the edit distance of the soundex codes

1. Keep the first letter (in uppercase).
2. Replace these letters with hyphens: a, e, i, o, u, y, h, w.
3. Replace the other letters by numbers as follows:
   - 1: b, f, p, v
   - 2: c, g, j, k, q, s, x, z
   - 3: d, t
   - 4: l
   - 5: m, n
   - 6: r
4. Delete adjacent repeats of a number.
5. Delete the hyphens.
6. Keep the first three numbers or pad out with zeros.

Source: Search Engines - IR in Practice, Croft et al. (p. 195)

Picking a spelling correction

A misspelled word can several possible corrections

```
lawers → lowers, lawyers, layers, lasers
```

```
trial lawers → trial lowers, trial lawyers, ...
```

Ranking of spelling corrections:

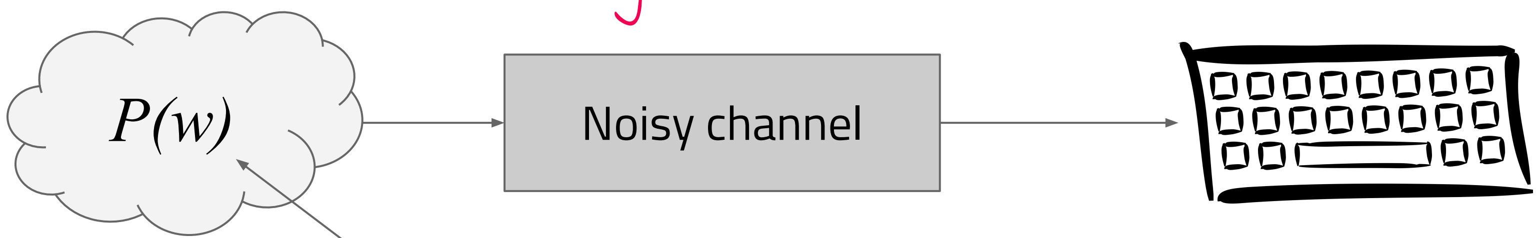
- Use of word frequency of occurrence in the language (**context** independent)
- Use of context and word frequencies leads to better results
- **Run-on errors**: word boundaries skipped or mistyped (whitespace can be treated as character)

Noisy channel model

user chooses to write w

brain as a noisy channel

user writes e instead of w



$P(e|w)$

language model
(with or without context information)

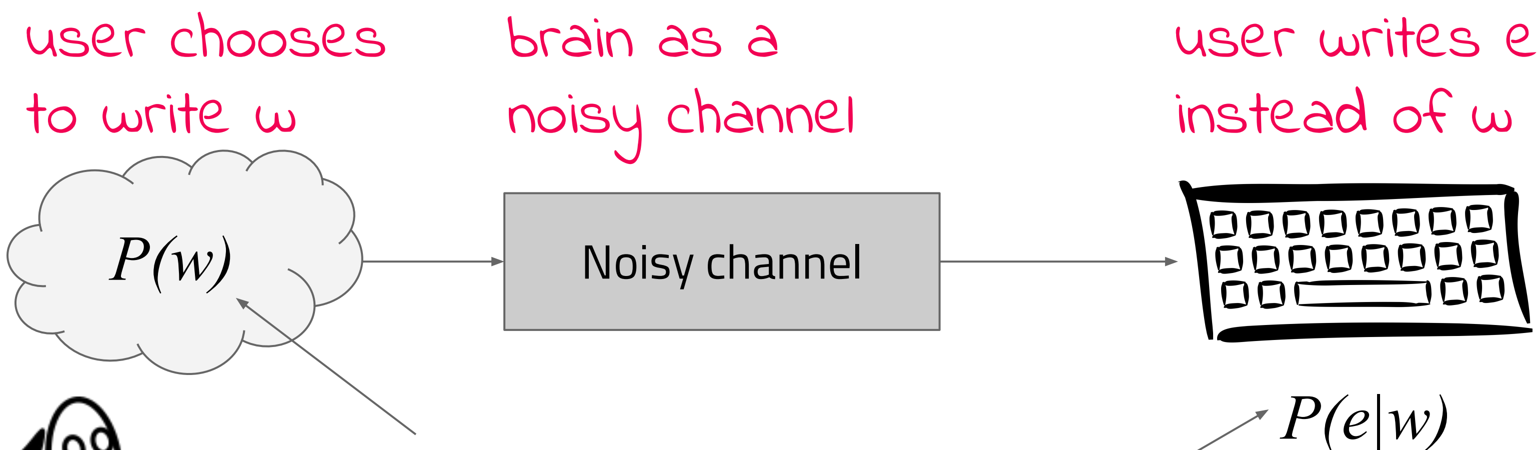
```
golf curse → golf course
```

error model
(information on the frequency of certain types of errors)

How do we estimate the probability $P(w|e)$?

$$P(w|e) \propto P(e|w) \times P(w)$$

Noisy channel model



language model
(with or without context information)

```
golf curse → golf course
```

error model
(information on the frequency of certain types of errors)

What about run-on errors and context?

probability that w follows w_p

$$P^{context}(w) = \lambda P(w) + (1-\lambda)P(w|w_p)$$

Source of probabilities

$P(w)$

- **Query log** (mostly for Web search), though frequency alone is not enough (e.g. britny spears)
- High-quality **document corpus** (e.g. news corpus)
- **Wikipedia history diffs** (small edits are often corrections)
- Trusted **lexicon**

$P(e|w)$

- Simple: all errors with the same edit distance have the same probability
- Complex: some errors are more likely than others, e.g. based on keyboard layout, source language, phonetics, cognitive misconceptions

Iterative spelling correction based on query logs

Recall or precision: which metric is more important for a Web search query spell checker?

1. Tokenize the query.
2. For each token, a set of alternative words and pairs of words is found using an edit distance modified by weighting certain types of errors, as described earlier. The data structure that is searched for the alternatives contains words and pairs from both the query log and the trusted dictionary.
3. The noisy channel model is then used to select the best correction.
4. The process of looking for alternatives and finding the best correction is repeated until no better correction is found.

albert einstein	4834
albert einstien	525
albert einstine	149
albert einsten	27
albert einsteins	25
albert einstain	11
albert einstin	10
albert eintein	9
albeart einstein	6
aolbert einstein	6
alber einstein	4
albert einseint	3
albert einsteirn	3
albert einsterin	3
albert eintien	3
alberto einstein	3
albrecht einstein	3
alvert einstein	3

Any string appearing in the query log can be a valid correction, even if misspelled.
The correct spellings tend to be more correct than the misspellings. Small mistakes are more common than large mistakes.

Iterative spelling correction based on query logs

Ablation study
remove some feature(s) and determine the system effectiveness compared to the compete setup.

Accuracy

	All queries	Valid	Misspelled
Nr. queries	1044	864	180
Full system	**81.8**	**84.8**	**67.2**
No lexicon	70.3	72.2	61.1
No query log	77.0	82.1	52.8
All edits equal	80.4	83.3	66.1
Unigrams only	54.7	57.4	41.7
1 iteration only	80.9	88.0	47.2
2 iterations only	81.3	84.4	66.7

https://www.microsoft.com/en-us/research/wp-content/uploads/2004/07/Cucerzan.pdf

Query autocompletion

NEXT WEEK

That’s it for query refinement!

Don’t forget that milestone M4 (March 19) **is coming up soon.**

Slack: in4325.slack.com

Email: in4325-ewi@tudelft.nl