THE UNIVERSITY *of* EDINBURGH

Text Technologies for Data Science

INFR11145

Query Expansion

Instructor:

Youssef Al Hariri

25-Oct-2023

Pre-Lecture

- How is progress on CW1 going?
  - There is no tricks!! ☺
- Test collection and queries:
  - Was announced on Monday 23 October 2023
  - NO questions related to CW1 are allowed anymore on Piazza till the deadline
    - Check existing questions and CW details. They cover everything.
  - Deadline: Friday 27 October 2023 – 12:00 PM (noon)
- Mid-year course feedback (link on Piazza)

Lecture Objectives

- Learn about Query Expansion
  - Query expansion methods
  - Relevance feedback in IR
  - Rocchio’s algorithm
  - PRF

- Implement:
  - PRF

Query Expansion

- Query: representation of user's information need
  - Many times it can be suboptimal
- Different words can have the same meaning
  - replacement, replace, replacing, replaced → Stemming
  - go, gone, went → Lemmatisation (NLP)
  - car, vehicle, automobile → ??
  - US, USA, the states, united states of America → ??
- Stemming/Lemmatisation → could be applied to normalise document and queries
  - Research shows that no significant difference between both
- Query Expansion (QE) → add more words of the same meaning to your query for better retrieval

Query Expansion: Methods

- Thesaurus
  - Group words into sets of synonyms (synsets)
  - Typically grouping is on the word level (neglects context)
  - Manually built: e.g. WordNet
    - NLTK wordnet: http://www.nltk.org/howto/wordnet.html
  - Automatically built:
    - Words co-occurence
    - Parallel corpus of translations
- Retrieved documents-based expansion
  - Relevance feedback
  - Pseudo (Blind) relevance feedback
- Query logs

Automatic Thesaurus: co-occurence

- Words co-occurring in a document/paragraph are likely to be (*in some sense*) similar or related in meaning
- Built using collection matrix (term-document matrix)
- For a collection matrix $\boldsymbol{A}$, where $\boldsymbol{A_{t,d}}$ is the normalised weight of term $\boldsymbol{t}$ in document $\boldsymbol{d}$, similarity matrix could be calculated as follows:

$$\boldsymbol{C = A.A^T}$$

where, $\boldsymbol{C_{u,v}}$ is the similarity score between terms $\boldsymbol{u}$ and $\boldsymbol{v}$. The higher the score, the more similar the terms

- Advantage: unsupervised
  Disadvantage: related words more than real synonyms

Automatic Thesaurus: co-occurence

- Example

Word	Nearest neighbors
absolutely	absurd, whatsoever, totally, exactly, nothing
bottomed	dip, copper, drops, topped, slide, trimmed
captivating	shimmer, stunningly, superbly, plucky, witty
doghouse	dog, porch, crawling, beside, downstairs
makeup	repellent, lotion, glossy, sunscreen, skin, gel
mediating	reconciliation, negotiate, case, conciliation
keeping	hoping, bring, wiping, could, some, would
lithographs	drawings, Picasso, Dali, sculptures, Gauguin
pathogens	toxins, bacteria, organisms, bacterial, parasite
senses	grasp, psyche, truly, clumsy, naive, innate

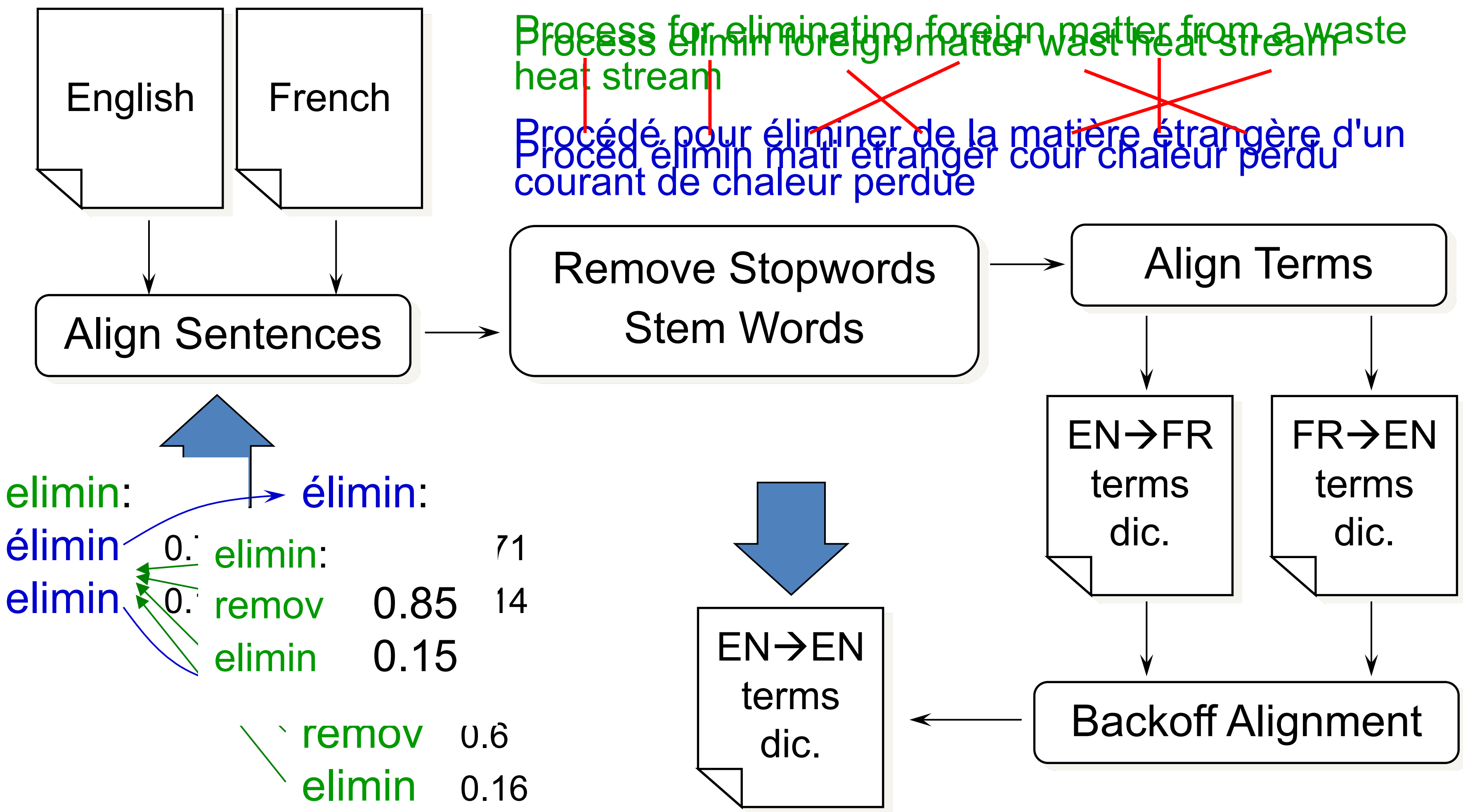
▶ **Figure 9.4** An example of an automatically generated thesaurus. This example is based on the work in Schütze (1998), which employs latent semantic indexing (see Chapter 18).

https://nlp.stanford.edu/IR-book/html/htmledition/automatic-thesaurus-generation-1.html#fig:autothesaurus

Automatic Thesaurus: parallel corpus

- Parallel corpus are the main training resource for machine translation systems
- Nature: sets of two parallel sentences in two different languages (source and target language)
- Idea:
  - More than one word in language X can be translated into the same word in language Y
    → these words in language X could be considered synsets
- Requirement: the presence of parallel corpus (training data) → supervised method

Automatic Thesaurus: parallel corpus



THE UNIVERSITY of EDINBURGH

Automatic Thesaurus: parallel corpus

- Example

motor	weight	travel	color	link
motor 0.63	weight 0.86	travel 0.67	color 0.56	link 0.4
engin 0.36	wt 0.14	move 0.19	colour 0.25	connect 0.18
		displac 0.14	dye 0.19	bond 0.17
				crosslink 0.13
				bind 0.12

cloth	tube	area	game	play
fabric 0.36	tube 0.88	area 0.4	set 0.6	set 0.3
cloth 0.3	pipe 0.12	zone 0.23	game 0.4	play 0.24
garment 0.2		region 0.2		read 0.17
tissu 0.14		surfac 0.17		game 0.16
				reproduc 0.1

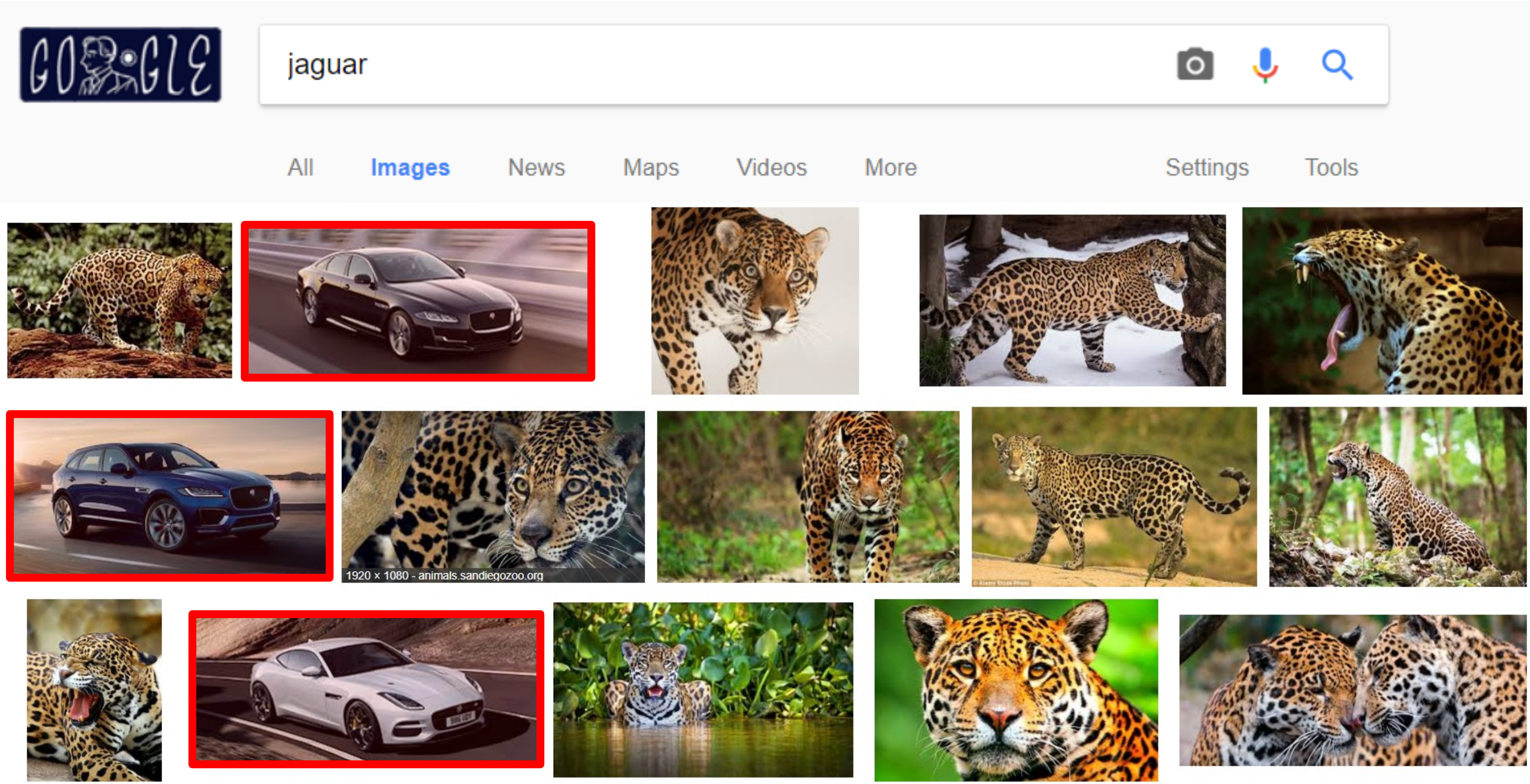
Thesaurus-based QE

- Works for very specific applications (e.g. medical domain)
- Many times fails to improve retrieval
  - Sometimes reduces both precision and recall
  - How?
- When it works, it is hard to get a consistent performance over all queries:
  - Improves some, and reduces others. Significant?
- Why it fails?
  - Lack of context
- Current research: word embeddings / BERT
  - No consistent improvement still

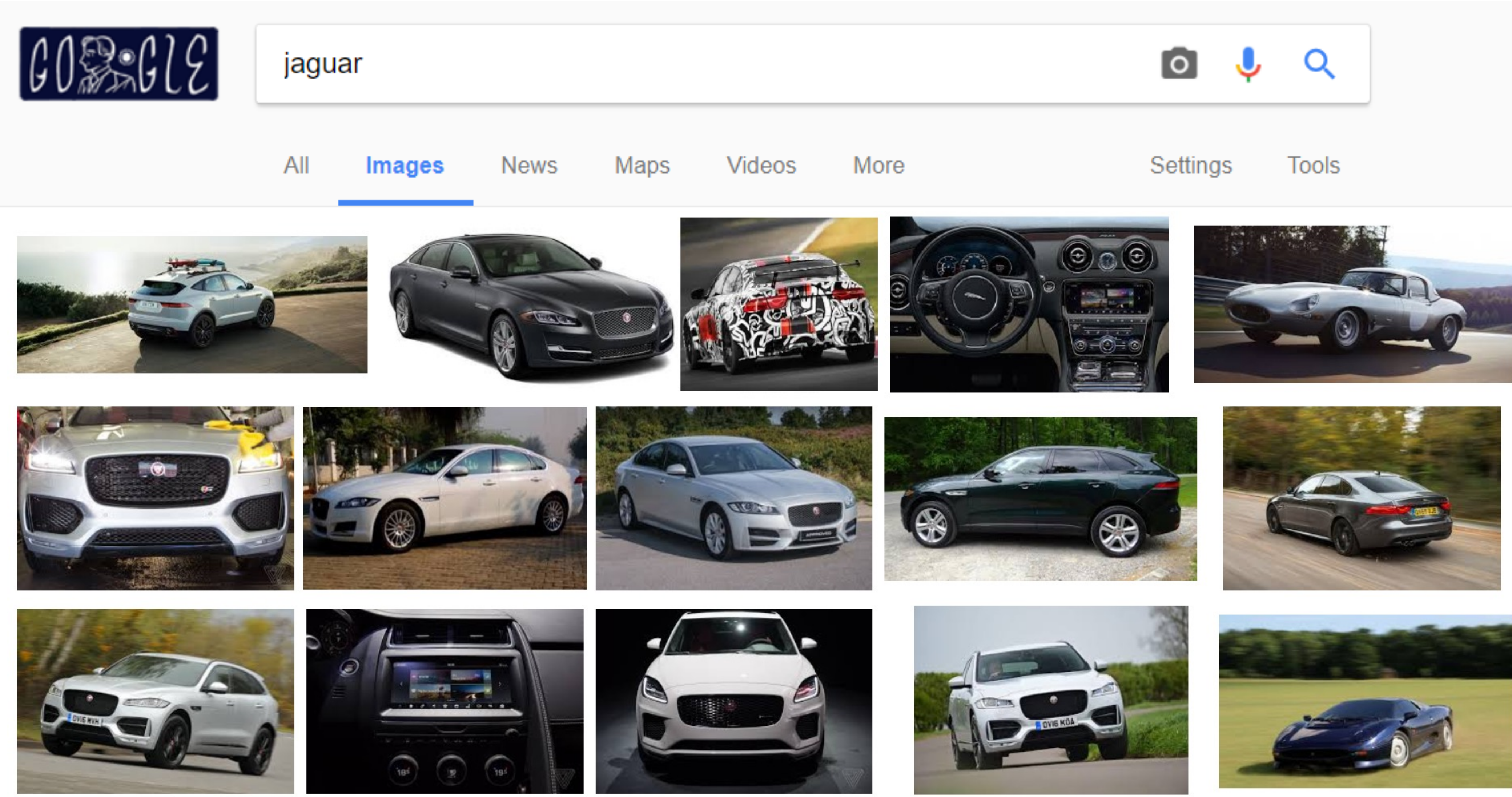
Relevance Feedback

- Idea: let user give feedback to the IR system about samples of what is relevant and what is not.
- User feedback on relevance of docs in initial results
  - User issues a (short, simple) query
  - The user marks some results as relevant or non-relevant.
  - The system computes a better representation of the information need based on feedback.
  - Relevance feedback can go through one or more iterations
- From user perspective: it may be difficult to formulate a good query when you don't know the collection well, BUT easier to judge particular documents

Example 1: Image Search



Example 1: Image Search



Example 2: Text Search

- Initial query: ***New space satellite applications***
- ***Initial Results***
  1. NASA Hasn't Scrapped Imaging Spectrometer
  2. NASA Scratches Environment Gear From Satellite Plan
  3. Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
  4. A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
  5. Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
  6. Report Provides Support for the Critics Of Using Big Satellites to Study Climate
  7. Arianespace Receives Satellite Launch Pact From Telesat Canada
  8. Telecommunications Tale of Two Companies
- User then marks relevant documents with "+"
- System learns new terms

New terms common in selected docs

2.074	new	15.10	space
30.81	satellite	5.660	application
5.991	**nasa**	5.196	**eos**
4.196	**launch**	3.972	**aster**
3.516	instrument	3.446	rianespace
3.004	bundespost	2.806	ss
2.790	rocket	2.053	scientist
2.003	broadcast	1.172	earth
0.836	oil	0.646	measure

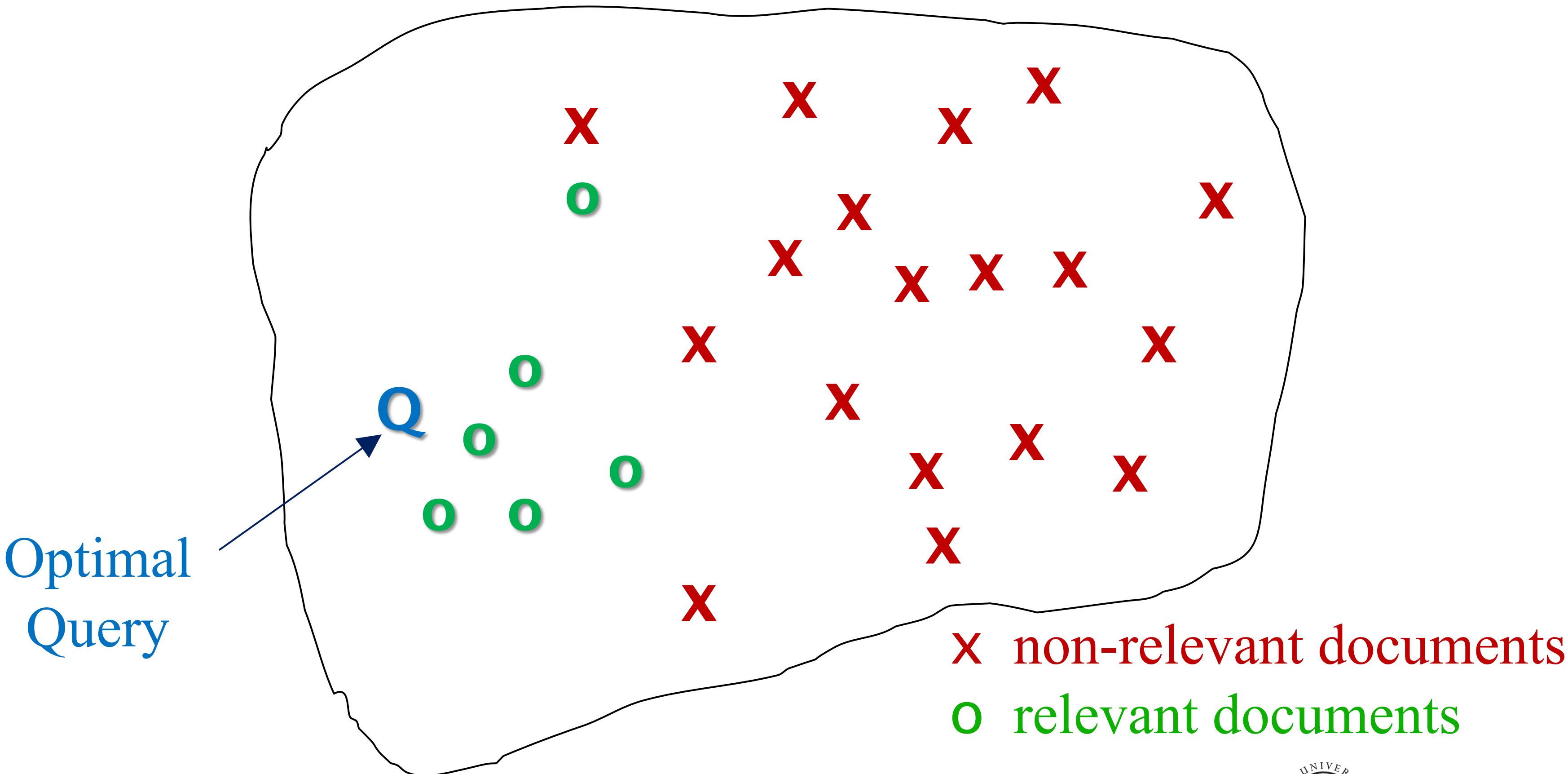
Adding new terms to the query

1. NASA Scratches Environment Gear From Satellite Plan
2. NASA Hasn't Scrapped Imaging Spectrometer
3. When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
4. NASA Uses 'Warm' Superconductors For Fast Circuit
5. Telecommunications Tale of Two Companies
6. Soviets May Adapt Parts of SS-20 Missile For Commercial Use
7. Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
8. Rescue of Satellite By Space Agency To Cost $90 Million

Hopefully better results!

Theoretical Optimal Query

- Found closer to *rel* docs and away from *irrel* ones.
- Challenge: we don't know the truly relevant docs



THE UNIVERSITY of EDINBURGH

Rocchio's Algorithm

- Key Concept: Vector Centroid
- Recall that, in VSM, we represent documents as points in a high-dimensional space
- The centroid is the centre mass of a set of points

$$\vec{\mu}(C) = \frac{1}{|C|} \sum_{\vec{d} \epsilon C} \vec{d}$$

where C is a set of documents.

- Introduced **1963**

Rocchio Algorithm: theory

- Rocchio seeks the query $\vec{q}_{opt}$ that maximizes

$$\vec{q}_{opt} = \underset{\vec{q}}{\operatorname{argmax}}[sim(\vec{q}, Crel) - sim(\vec{q}, Cirrel)]$$

- For Cosine similarity

$$\vec{q}_{opt} = \frac{1}{|Crel|} \sum_{\vec{d_j} \in C_{rel}} \vec{d_j} - \frac{1}{|C_{irrel}|} \sum_{\vec{d_j} \notin C_{rel}} \vec{d_j}$$

$$\vec{q}_{opt} = \vec{\mu}(C_{rel}) - \vec{\mu}(C_{irrel})$$

Rocchio Algorithm: in practice

- Only small set of docs are known to be *rel* or *irrel*

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_{rel}|} \sum_{\vec{d}_j \in D_{rel}} \vec{d}_j - \gamma \frac{1}{|D_{irrel}|} \sum_{\vec{d}_j \in D_{irrel}} \vec{d}_j$$

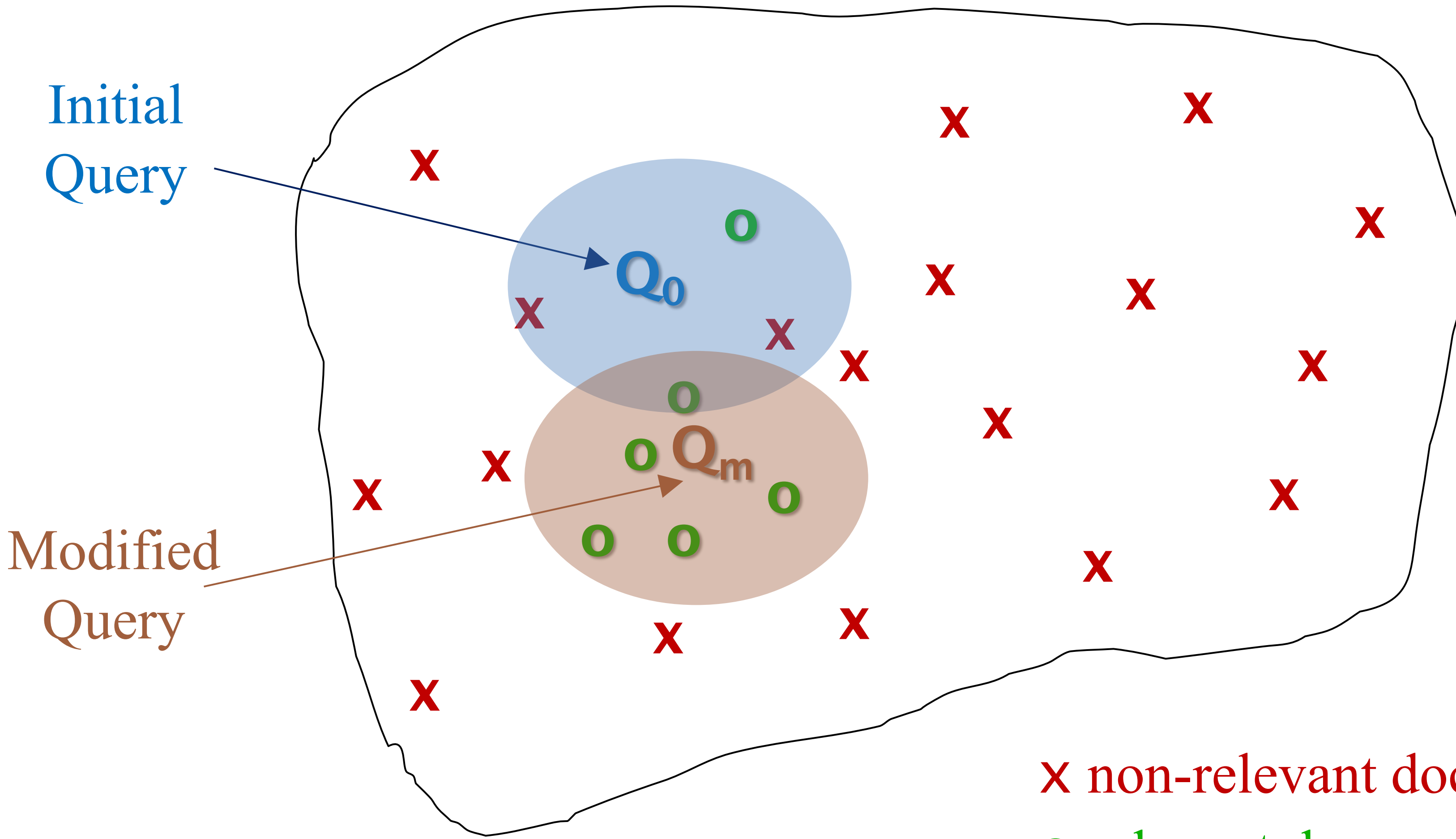
$\vec{q}_0$ = original query vector
D_{rel} = set of known relevant doc vectors
D_{irrel} = set of known non-relevant doc vectors
$\vec{q}_m$ = modified query vector
α = original query weights (hand-chosen or set empirically)
β = positive feedback weight
γ = negative feedback weight

- New query moves toward relevant documents and away from non-relevant documents

Notes about setting weights: α, β, γ

- Values of β, γ compared to α are set high when large judged documents are available.
- In practice, +ve feedback is more valuable than -ve feedback (usually, set β>γ)
  - Many systems only allow positive feedback (γ=0).
  - Or, use only highest-ranked negative document.
- When γ>0, some weights in query vector can go -ve.
  - "Jaguar" $\xrightarrow{feedback}$ jaguar + car + model - animal - jungle
- In practice, top n_t terms in $\overrightarrow{d_j} \in Drel$ are only selected
  - n = 5 → 50
  - Top n_t are identified using e.g. TFIDF

Effect of Relevance Feedback on Query



x non-relevant documents
o relevant documents

Effect of Relevance Feedback on Retrieval

- Relevance feedback can improve recall and precision
- In practice, relevance feedback is most useful for increasing recall in situations where recall is important.
- Empirically, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.

Relevance Feedback: Issues

- Long queries are inefficient for typical IR engine.
  - High cost for retrieval system. (why?)
  - Long response times for user.
- It’s often harder to understand why a particular document was retrieved after applying relevance feedback
- Users are often reluctant to provide explicit feedback → not practical!

Relevance Feedback: Practicality

- User revises and resubmits query
  - Users may prefer revision/resubmission to having to judge relevance of documents.
  - Useful for query suggestion to other users

- Is there a way to apply relevance feedback without user's input?

Pseudo (Blind) Relevance Feedback

- Solves the problem of users hate to provide feedback
- Feedback is applied blindly (PRF)
  - Automates the “manual” part of true relevance feedback.
- Algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - Assume that the top *k* documents are relevant
  - Do relevance feedback (e.g. Rocchio)
  - Typically applies only positive relevance feedback (γ=0)
- Mostly works
  - Still can go horribly wrong for some queries (when top *k* docs are not relevant)
  - Several iterations can lead to query drift

PRF (BRF)

- Was proven to be useful for many IR applications
  - News search (learn names and entities)
  - Social media search (learn hashtags)
  - Web search (implicit feedback is used more = clicks)
- Some domains are more challenging
  - Patent search
    - Top documents are usually not relevant
    - Patent text in general is unclear/confusing
- PRF is the most basic QE method for IR
  - Unsupervised
  - Language independent
  - Does not require any kind of language resources

PRF (BRF): Evaluation

- In practice, different number of feedback docs (n_d) and terms (n_t) are usually tested for PRF
  - n_d: 1 → 50
  - n_t: 5 → 50
- Results of PRF are directly compared to baseline (with no PRF)
  - It is <u>not</u> considered cheating.
  - It is essential to show that improvement is significant, and preferred to show the % of queries improved vs degraded.

Practical

Summary

- QE: automatically add more terms to user's query to better match relevant docs
- QE via thesaurus
  - Manual/automatic thesaurus: useful for specific applications
  - Fail when context is important
- Relevance feedback
  - Get samples of *rel*/*irrel* docs for extracting QE useful terms
  - Rocchio's is one of the most common algorithms for query modification
- PRF
  - Skips user's input for the feedback process
  - Found to be useful in many applications

Resources

- Text book 1: Intro to IR, Chapter 9
- Text book 2: IR in Practice, Chapter 6.2, 6.3
- Reading:
  Magdy W. and G. J. F. Jones.
  A Study on Query Expansion Methods for Patent Retrieval.
  PAIR 2011 - CIKM 2011 (link)
- Lab 5