Big Data and Data Mining

Advanced methods

Flavio Bertini

flavio.bertini@unipr.it

Synonymy

- In most collections, the same concept may be referred to using **different words**
- This issue, known as **synonymy**, has an impact on the **recall** of most information retrieval systems
  - For example, you would want a search for **aircraft** to also match the word **airplane**

Query expansion (1/3)

- Idea: we augment the query with keywords synonyms

User Query: “car”

Expanded Query: “car cars automobile automobiles auto”

Query expansion (2/3)

- Idea: we augment the query with keywords synonyms and related terms
- A variety of automatic or semi-automatic query expansion techniques have been developed
  - goal is to improve effectiveness by matching related terms
  - semi-automatic techniques require user interaction to select best expansion terms
- Query suggestion is a related technique
  - alternative queries, not necessarily more terms

Query expansion (3/3)

- Query expansion involves techniques such as:
  - Finding synonyms of words
  - Finding semantically related words
  - Finding all the various morphological forms of words by stemming each word in the search query
  - Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results
  - Re-weighting the terms in the original query

Related terms

- Where to find terms related to a query, in order to expand it?
  - Controlled vocabularies
    - **WordNet** - *A Lexical Database for English* [Princeton University]
  - Text collection
    - **Co-occuring** terms
    - Terms from relevant documents
    - Terms from retrieved documents
    - Terms in an adjacent **window** (of relevant or retrieved documents)

Thesaurus query expansion

- Automatic expansion based on general controlled vocabulary (thesaurus) is **not much effective**
  - It does not take **context** into account:

Query: “tropical fish <u>tanks</u>”

Expanded query: “tropical fish tanks <u>aquariums</u>”

Query: “armor for <u>tanks</u>”

Expanded query: “armor for tanks <u>aquariums</u>”

Co-occurence query expansion

- Instead of using a thesaurus, related keyword can be extracted from text collections
- Different measures of **co-occurrence** can be used to find related keywords:
  - Dice's coefficient
  - Mutual information
  - Expected mutual information
  - Pearson's Chi-squared (χ^2)
- Measure are based on **entire documents** or smaller **parts of documents** (sentences, paragraphs, windows). We will consider entire documents now, for simplicity

Dice’s coefficient (1/2)

- Suppose we want to find words related to “fish”
- How to measure the “relatedness” of a second term to the word fish?
- A measure of co-occurrence:

$$\frac{\text{How many times they appear } \textbf{together}}{\text{How many times they appear } \textbf{singularly}}$$

- Idea: the higher this score, the more related should be the two words!

Dice's coefficient (2/2)

- Term association measure used since the earliest studies of **term similarity** and **automatic thesaurus construction** in the 1960s and 1970s
- Given two words **a** and **b**, it is formally defined as:

$$2n_{ab}/(n_a + n_b)$$

  - $\mathbf{n_{ab}}$ is the number of documents containing **both** words **a** and **b**
  - $\mathbf{n_a}$ is the number of documents containing word **a**
  - $\mathbf{n_b}$ is the number of documents containing word **b**

Mutual information

- It has been used in a number of studies of word collocation
- Similar to Dice, based on probabilities
- For two words **a** and **b**, it is defined as

$$\log \frac{P(a,b)}{P(a)P(b)}$$

  - **P(a, b)** is the probability that a and b occur in the same text window
  - **P(a)** is the probability that word **a** occurs in a text window
  - **P(b)** is the probability that word **b** occurs in a text window

Mutual information: problem

- A problem with mutual information is that it tends to favor low-frequency terms
- For example:
  - Consider two words **a** and **b**:
    - $\mathbf{n_a = n_b}$= 10 and co-occur **half the time** n_{ab} = 5
    - Mutual information for these two terms is $\mathbf{5 \cdot 10^{-2}}$
  - Consider two words **c** and **d**:
    - $\mathbf{n_c = n_d}$= 1000 and co-occur **half the time** n_{cd} = 500
    - Mutual information for these two terms is $\mathbf{5 \cdot 10^{-4}}$
- **Both pairs co-occur half of the time they occur.** However, they have different mutual information: 0.05 vs 0.0005

Expected mutual information

- The *expected mutual information* addresses the **low-frequency problem** by weighting the mutual information value using the probability P(a,b)
- We are primarily interested in the case where both terms occur, giving the formula:

$$P(a,b) \cdot log \frac{P(a,b)}{P(a)P(b)}$$

- In the previous case
  - $n_{ab} = 5$, $MI_{ab} = 5 \cdot 10^{-2} \rightarrow eMI_{ab} = 0.25$
  - $n_{cd} = 500$, $MI_{cd} = 5 \cdot 10^{-4} \rightarrow eMI_{cd} = 0.25$

Pearson's Chi-squared (χ^2)

- This measure
  - compares the number of co-occurrences of two words with the expected number of co-occurrences if the two words were independent
  - normalizes this comparison by the expected number

$$\frac{(n_{ab} - N \cdot \frac{n_a}{N} \cdot \frac{n_b}{N})^2}{N \cdot \frac{n_a}{N} \cdot \frac{n_b}{N}}$$

- N is the number of documents in a collection
- $N \cdot \frac{n_a}{N} \cdot \frac{n_b}{N}$ is the expected number of co-occurrences if the two terms occur independently

Query expansion: an example

- Using a **TREC news collection** the four co-occurrence measures are applied on a document level
- Top-5 related words are shown
- Word for which we are searching related terms is

fish

Query expansion results

Dice's coefficient	Mutual information	Expected mutual information	Pearson's Chi-squared
species	zoologico	water	arslq
wildlife	zapanta	species	happyman
fishery	wrint	wildlife	outerlimit
water	wpfmc	fishery	sportk
fisherman	wighout	sea	lingcod

- Mutual information favors very rare words (sometimes mistyped words!)
- Chi-squared also capture unusual words
- Dice's coefficient and Expected mutual information are more suitable for IR query expansion

Query expansion with relevance feedback

- Relevance feedback (RF) is a query expansion and refinement technique based on **user feedback**
- General idea:
  1. The user issues a (short, simple) query
  2. The system returns an initial set of results
  3. The user marks some returned documents as relevant (or non relevant)
  4. The system computes a better representation of the information need based on the user feedback
  5. The system displays a revised set of retrieval results

RF example (1/2)

Query: *New space satellite applications*		
Rank	Document Title	User Feedback
1	**NASA Hasn’t Scrapped Imaging Spectrometer**	**YES**
2	**NASA Scratches Environment Gear From Satellite Plan**	**YES**
3	Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes	NO
4	A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget	NO
5	Scientist Who Exposed Global Warming Proposes Satellites for Climate Research	NO
6	Report Provides Support for the Critics Of Using Big Satellites to Study Climate	NO
7	Arianespace Receives Satellite Launch Pact From Telesat Canada	NO
8	**Telecommunications Tale of Two Companies**	**YES**

RF example (2/2)

Query: *New space satellite applications*

Documents relevant
from the user feedback

#1	**NASA Hasn't Scrapped Imaging Spectrometer**
#2	**NASA Scratches Environment Gear From Satellite Plan**
#8	**Telecommunications Tale of Two Companies**

Recurring keywords in
relevant documents

new, space, satellite,
application,
**nasa, eos, launch,
aster, instrument,
arianespace,
bundespost, ss,
rocket, scientist,
broadcast, earth,
oil, measure**

Expanded query:

new space satellite application + ***nasa eos launch aster instrument arianespace bundespost ss rocket scientist broadcast earth oil measure***

Pseudo RF

- *Pseudo relevance feedback*, also known as *blind relevance feedback*, provide a method for **automatic** relevance feedback
- It automates the manual part of RF, so that the user gets improved retrieval performance **without an extended interaction**
- The method involves the following:
  1. normal retrieval to find an initial set of most relevant documents
  2. assume that the **top k** ranked documents are **relevant**
  3. compute RF as before under this assumption

Machine Learning and IR: Why?

- Suppose we want to consider (**combining**) at the same time:
  - term frequency in the document body
  - term frequency in the document title
  - document length
  - document popularity (e.g. PageRank)
- ...as “features” to estimate the relevance, how we should **weight** each feature?
- A **learning to rank** model learn the weights from a training set of features and relevance judgements

Machine Learning and IR

- Idea: using machine learning (ML) to build a classifier that classify documents into **relevant and non-relevant classes**
- Although ML has been around for a long time, this good idea has been researched only recently
  - **Limited training data**: it was very hard to gather test collection queries and relevance judgments that are representative of real user needs
  - Traditional ranking functions in IR used a very **small number of features**:
    - Term frequency
    - Inverse document frequency
    - Document length

Learning to rank

- In the last 10 years things have changed
- Modern systems – especially on the Web – use a **great number of features**
  - Log frequency of query word in anchor text
  - Query word color on page
  - # of images on page
  - # of (in/out) links on page
  - PageRank of page
  - URL length
  - URL contains query terms
  - Page edit recency
  - Page length
  - ...
- Lot of **training data** is available from huge query logs that are collected from user interactions

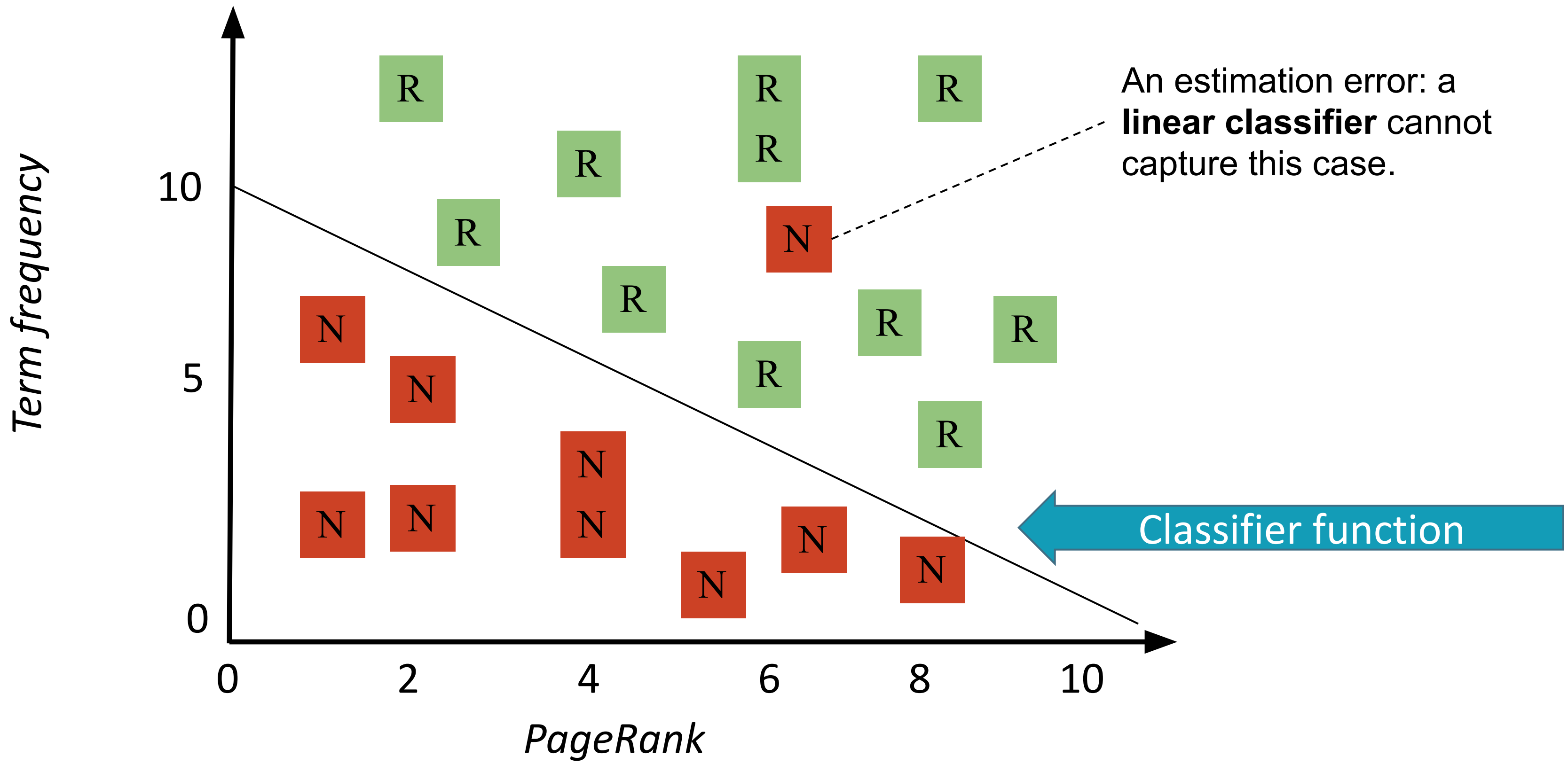
Example: features

- Suppose we are considering two features in each document:
  - Term frequency ***tf***: how many times the query terms are found in the document
  - Pagerank ***pr***: popularity of the document
- Training set is made of (tf,pr) vectors each with a correspondent relevance judgment
- A learning to rank model given the document features as input (tf,pr), should output the estimated relevance

Example: training data

Query: “*fish tank price*”		INPUT		OUTPUT
Training sample	**Doc ID**	**Term frequency**	**PageRank**	**judgement**
001	37	11	3	1 (relevant)
002	38	0	8	0 (non-relevant)
003	238	8	2	1 (relevant)
004	248	1	2	0 (non-relevant)
005	1741	5	6	1 (relevant)
006	2094	18	1	1 (relevant)
…	…	…	…	…

Example: classifier



- The learned **weights** are the **coefficients** of a linear function
- The function represent the learned model that **separates** relevant (output 1) from non relevant (output 0) documents based on the two input variables

Relevance feedback and learning

- RF is a simple example of using supervised machine learning in information retrieval: **training data** (i.e., the identified relevant and non-relevant documents) is used to improve the system's performance
- In the last example, we have used the relevance judgements of a test collection to train a classifier: this is called **offline learning**
- Using relevance feedback to tune the classifier weights and improve its accuracy is an example of **online learning**

References

[Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze
Introduction to Information Retrieval
Cambridge University Press. 2008]

The book is also online for free:

- HTML edition (2009.04.07)
- PDF of the book for online viewing (with nice hyperlink features, 2009.04.01)
- PDF of the book for printing (2009.04.01)