Multimedia Technology

Lecture 2 Document Retrieval: the model

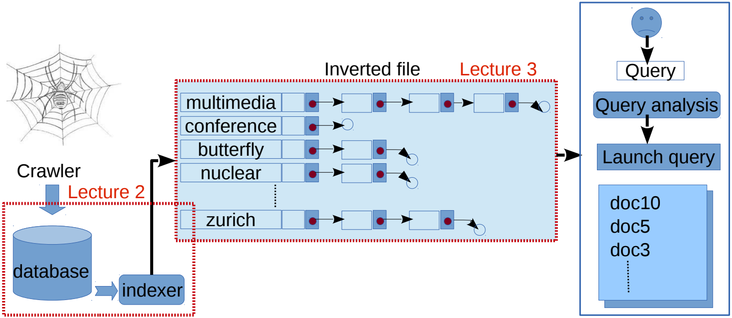
Lecturer: *Dr*. Wan-Lei Zhao

Autumn Semester 2024

Email: wlzhao@xmu.edu.cn,

Outline

IR framework: recap



- Focus on pre-processing steps on text documents
- Take mainly English documents as examples

Human Languages (1)

- 7,000 languages in the world
- 90% of these languages are used by less than 100,000 people
- Based on your knowledge and imagination
- Please list out top-5 most popularly used languages
- Give the rank also, do it now ...

Human Languages (2)

- 7,000 languages in the world
- 90% of these languages are used by less than 100,000 people

Language	Population	Category	Region
Mandarin	1.2 billion	isolating language	China
English	508 million	inflectional language	UK, North America
Hindi	497 million	inflectional language	India & Pakistan
Spanish	392 million	inflectional language	Span & South America
Russian	277 million	inflectional language	Russia & East Europe

我昨天去图书馆借了三本书

I borrowed three books from the library yesterday

- **Mainly** talk about retrieval on English documents
- Mention **a little** about processing on Chinese documents

Human Languages (3)

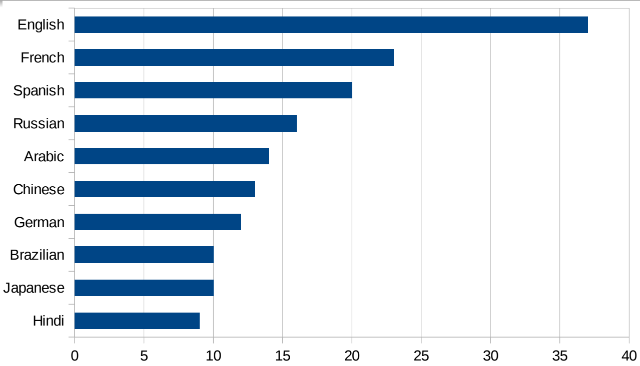
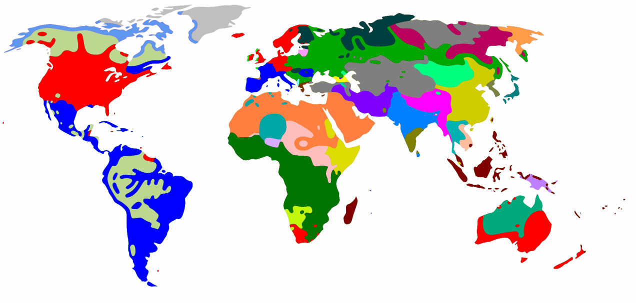


Figure: Weights of real impact to the world.

- In terms of real influence, the rank changes[1]
- Influence: economically, politically, size of population and number of countries

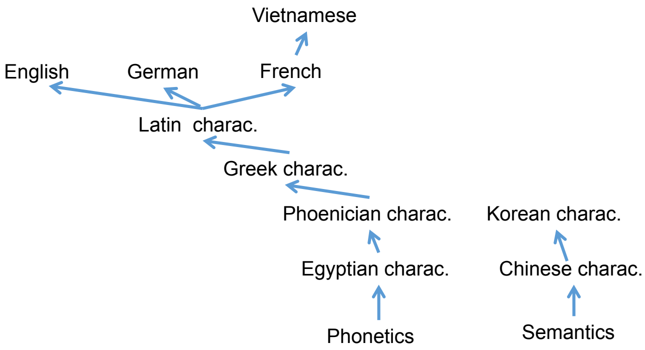
[1] Conducted by Webb.

Distribution of World Languages



- Pay attention that not all the languages have their written forms

How many Language characters in this World



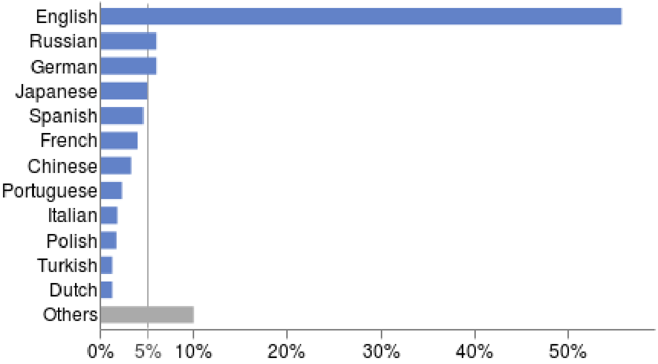
- In summary, there are only two types of characters in this world

Acient Egyptian	Chinese	Phoenician	Greek	English
	牛\牛头		α	A
	房子		β	B
	骆驼		γ	C
	门\鱼		Δ	D
	窗		ε	E
	钩		υ	F
	武器		ζ	G

Parsing a document

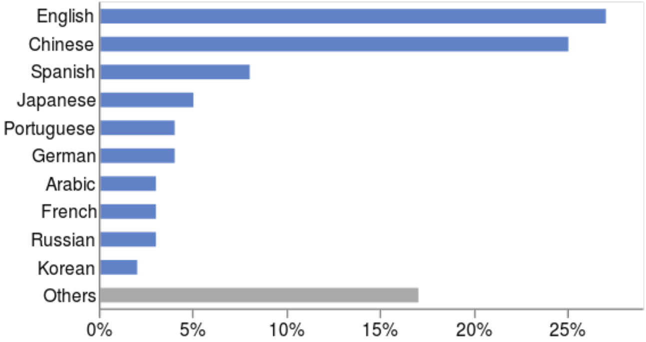
- Popular document formats
  - html
  - pdf/ps
  - doc/docx/odt/rtf
- Different codes
  - UTF-8
  - CP1252
  - GB22238-1008
- Different languages
  - English ($>$ 55%)
  - Russian ($>$ 5%)
  - German ($>$ 5%)
  - Chinese ($<$ 5%)
- All are modelled as classification problem
- In practice, they are handled in a heuristic way

Documents in the web (1)



- More than 55% of the websites are in English
- Google supports most of the languages
- In most of the countries (except China), Google is the first option
- Statistics are conducted in 2011

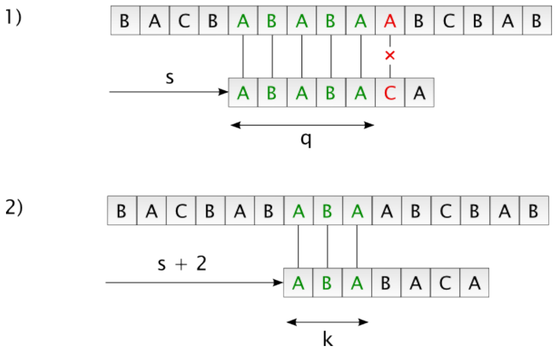
Documents in the web (2)



- Size of Internet users matters
- Another wave of booming in China is still expected
- Advertisement is major income for most of the search engines

A naive solution for documents retrieval: KMP (1)

Prefix function



- KMP proposed by Knuth, Morris and Pratt
- Linear complexity for string matching

A naive solution for documents retrieval: KMP (2)

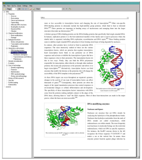
- IR can be modelled as a string matching problem
  - Given a query string
  - KMP matches it against the whole corpus
  - KMP returns all the locations that the query string occurs
- Does it work? And why?
- **Two minutes** to think about it

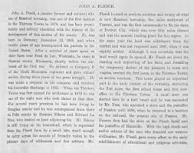
A naive solution for documents retrieval: KMP (3)

- Think about the size of corpus
- Think about the time cost in the worst case
- Users only accept less than 1 second delay
- It is not error tolerant
- KMP is still popular in Bio-informatics

Document: the basic indexing unit

- People compose, share and keep information in document granularity
- Aim of retrieval is to find relevant documents/books
- Documents are presented in different forms
  - One piece of email, web page, pdf docs etc.
  - Slides of one presentation
  - Scanned documents (Google book)





- Bunch of tools to convert them into pure text documents

Docs to words: articles, propositions and etc.

- Given words are extracted from a piece of BBC News
- Try to figure out what the news is about

article, adverb, pronoun, preposition and conjunction

A ... the ... will ... not ... as ... as ... to ...
... the ... for ... an ... in ... that ... he ... the ... on the ...
The ... in ... the ... as ... if ... the ... of ... by the ...
The ... of ... of ... of ... of the ... of ... in ... to the ... it
... to ... the ... from ... to ... and ... of the... but ... since ... to

- Safe to say, articles and propositions are not helpful

Docs to words: adverbs and adjectives

adjective and adverb

costly ... previously ... effective ...
… chief ... economic ...
economic ... high ... significantly … hardest ...
highest ... economic ... great ... still ...
… largely successful ... small ...

- Safe to say, adverbs and adjectives are not helpful either

Docs to words: verbs

- How about verbs?

says ... be ... feared ... thanks ...

… told ... expect ... range ...

had predicted ... impact ... could be … spread ...

been reduced ... said ... gone ... needed ...

… said ... keep ... were ... declared ... seeking ... contain …

- Safe to say, verbs are not helpful either

Docs to words: verbs, adverbs and adjectives

- How about verbs, adverbs and adjectives?

verb, adjective and adverb

says … costly ... previously feared … thanks ... effective ...

… chief ... told … economic ...

had predicted … economic … could be high ... spread significantly … hardest ...

highest … been reduced … said … economic … gone ... great ... still ...

… said … keep … were largely successful … declared ... seeking … contain … small ...

- More meaningful, however do not work again!

Docs to words: nouns

world bank official ... Ebola epidemic ... west Africa economy … efforts

Francisco Ferreira … bank ... economist ... Africa ... audience ... Johannesburg … Wednesday … toll … region … billion

World bank ... October ... impact … billion … virus … border … Guinea Liberia … Sierra Leone … countries … hit … outbreak

… risk ... case ... impact ... Ebola ... success … containment … countries … Ferreira … Reuters news agency … level … preparedness … focus

Ferreira … efforts ... outbreak ... spreading ... countries … Senegal … Nigeria ... cases … diseases … outbreak

- What is your observation?
- Outline what the news is about

Tokenization

- The process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements
- Elements are called tokens
- Input:
  - Jim and his wife visited Golden Gate Bridge in San Francisco
- Output (tokens):
  - Jim
  - and
  - his
  - wife
  - visited
  - Golden Gate Bridge
  - in
  - San Francisco

Tokenization: recognize special terms/phrases

- Maintain a corpus
- Update the corpus from time to time
- Internet culture is always changing
- Few examples
  - “no zuo, no die”
  - “guanxi”
  - “SIFT”
  - “CNN”

Tokenization: numbers

- Numbers are ignored in old IR systems
- They are useful in many ways
  - E.g. “911” means different things in different contexts
  - “3.1415926” means “π”
  - People or institutes can be localized by the phone numbers or post code
  - Input post code “361005” with Google, see what happens

Tokenization: language issues (1)

- In most of the reflecting languages, words are separated by spaces
- English is a good example, however not always true
- Guess what the German sentence is about
    - Lebensversicherungsgesellschaft Mitarbeiter

Tokenization: language issues (2)

- In most of the reflecting languages, words are separated by spaces
- English is a good example, however not always true
- Guess what the German sentence is about
    - Lebensversicherungsgesellschaft Mitarbeiter
    - "life insurance company employee"
    - Compound word splitter is required and very helpful
    - Similar case for languages such as Chinese and Japanese
- Arabic is another case, reads from right to left
- Numbers are from left to right, words are separated by spaces

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

- Many softwares handle the tokenization[2]

[2] http://nlp.stanford.edu/software/tokenizer.shtml

Stop words

- The most common words
  - “the”, “a”, “there”, “be” ...
  - Similar case for Chinese texts
- Remove words according to a stop words list
- Exceptions
  - “to be, or not to be”
  - “Alexander the Great”
  - “state of the art”

Case-folding, Normalization and spelling correction

- Case folding: reduce all the letters to lower case
- For most of users, they ignore proper capitalizing
- Normalization: normalize different writings of one term into one
    - "Windows", "Window" to "windows"
    - "colour" to "color"
    - "fig." to "figure"
    - E.g. "U.S.A", "united states" to "usa"
- Correcting spelling mistakes
    - "stastics" to "statistics"
    - "questionairs" to "questionaires"
    - Done by searching for closest words (Hamming distance)

Stemming

- Reduce words into their roots before indexing
- E.g. "indexing", "indices" to "index"
- E.g. "automatic", "automatically", "automate" to "automat"
  - Available codes: Porter, Lovins and Snow ball etc.
  - It is language dependent
  - No stemming is needed for isolating languages

Review on the pre-processing steps

George Bush watched movis about "9-11" in San Fransisco

Tokenize → George Bush | watched | movis | about | 9-11 | in | San Fransisco

Stop-words → George Bush | watched | movis | 9-11 | San Fransisco

Normalize Spelling correct → george bush | watched | movies | 9-11 | san fransisco

Group nouns → george bush | movies | 9-11 | san fransisco

Stemming → george bush | movie | 9-11 | san fransisco

Outline

Boolean model (1)

- Simple model based on **set theory** and **boolean algebra**
- Queries specified as boolean expressions
- Document is represented as a binary vector, only indicates whether a term appears
  - Quite intuitive
  - Allows the query to be expressed in precise semantics (1st order)
  - Neat formalism
- Given a mini vocabulary $V = \{document, retrieve, multimedia, class\}$
- A mini document $D = \{multimedia, class\}$
- Document D is represented as d=[0,0,1,1]

Boolean model (2)

Table: Boolean representation of four documents

	religion	sun	moon	earth	nicolaus copernicus
	t_1	t_2	t_3	t_4	t_5
d_1	1	1	0	1	1
d_2	0	1	1	1	0
d_3	0	1	0	0	1
d_4	1	0	0	0	0

- Given query Q={sun, nicolaus copernicus}
- Relevant documents are d_1 and d_3
- However, we have no idea which one is more similar to the query

Boolean model (3): Jaccard distance

Table: Boolean representation of four documents

	religion	sun	moon	earth	nicolaus copernicus
	t_1	t_2	t_3	t_4	t_5
d_1	1	1	0	1	1
d_2	0	1	1	1	0
d_3	0	1	0	0	1
d_4	1	0	0	0	0

- Given query Q={sun, nicolaus copernicus}
- Given document d_i is represented as a set of terms

$$Sim(Q, d_i) = \frac{|Q \cap d_i|}{|Q \cup d_i|} \tag{1}$$

- **Jaccard** dist. is able to rank the retrieved documents according to $Sim(Q, d_i)$

Boolean model (4): complicated query

- Given query: earth and moon or earth without sun
- $\{t_4 \wedge t_3 \vee t_1\}$
- Can be expressed in disjunctive normal form (DNF)

$$
\begin{aligned}
q = & (0 \wedge 0 \wedge 1 \wedge 1 \wedge 0) \vee \\
& (0 \wedge 0 \wedge 1 \wedge 1 \wedge 1) \vee \\
& (1 \wedge 0 \wedge 1 \wedge 1 \wedge 0) \vee \\
& (1 \wedge 0 \wedge 1 \wedge 1 \wedge 1) \vee \\
& (0 \wedge 1 \wedge 1 \wedge 1 \wedge 0) \vee \\
& (0 \wedge 1 \wedge 1 \wedge 1 \wedge 1) \vee \\
& \qquad\qquad\quad . \\
& \qquad\qquad\quad .
\end{aligned} \tag{2}
$$

Boolean model (5): advantages and disadvantages

- Advantages:
  - Good ability in expressing complicated retrieval request
- Disadvantages:
  - One cannot expect every user express their request in boolean expression smoothly
  - Different words should have different weighting
  - E.g. "nicolaus copernicus" should be given higher weight than the rest
  - Does not support partial matching
  - How about document only contains term "earth"?

Vector model (1): representation

- Document is represented as a vector
- One term t_i is associated with a weight

$$d_j = \{w_{1j}, w_{2j}, w_{3j} \dots w_{ij} \dots w_{nj}\}$$

- Advantages
  - Term is weighted according to its importance
  - d_j is usually a **SPARSE** vector
  - Supports partial Matching
- Query is represented as

$$q = \{w_{1q}, w_{2q}, w_{3q} \dots w_{iq} \dots w_{nq}\}$$

Term weighting (1)

- The terms in a document are not equally useful for describing the document contents
- In the previous example, "nicolaus copernicus" is specific term
- Intuitively, documents contain "nicolaus copernicus" are highly relevant to query that also contains this term
- In contrast, term appears in every document is less useful
- That is why we use "stop words" list in the pre-processing

Term weighting (2)

- Each term in document d_j is associated with a weight w_{ij}
- If term t_i does not occur in d_j, $w_{ij} = 0$
- w_{ij} is the number of occurrences that term i in d_j

Term	Frequency
earth	1
mercury	2
planet	1
sun	1

Both Earth and Mercury are planet. Mercury is the one closest to Sun.

Term	Frequency
nicolaus copernicus	1
astronomer	1
sun	1
earth	1
center	1

Nicolaus Copernicus is an astronomer. He found that the Sun rather than the Earth at its center.

- The number of terms in a document ranges from several dozens to a few hundreds
- Vector $d_j = \{w_{1j}, w_{2j} \ldots w_{nj}\}$ is very sparse; Vocabulary size is usually larger than $10,000$
- Inverted files (discussed in later lectures) becomes very helpful

Term	Frequency
earth	1
mercury	2
planet	1
sun	1

Both Earth and Mercury are planet. Mercury is the one closest to Sun.

Term	Frequency
nicolaus copernicus	1
astronomer	1
sun	1
earth	1
center	1

Nicolaus Copernicus is an astronomer. He found that the Sun rather than the Earth at its center.

Term weighting (3)

- Frequency of term t_i in the corpus
- Total number of occurrences of term t_i in the corpus

$$F(t_i) = \sum_j w_{ij} \tag{3}$$

- Document frequency of term t_i:
- The number of documents that term t_i occurs

$$DF(t_i) = \sum_j t_i \in d_j \tag{4}$$

- It is obvious $DF_i \leq F_i$

Term weighting (4): example

$$F(planet)=2+2+1$$

$$F(sun)=1+2+1$$

Earth is the third planet from Sun. It is the densest Planet. It is only planet that so far we know life exists.

Mercury is the smallest and the closest planet to the Sun. It orbits around the Sun faster than any other planets.

$$DF(planet)=1+1+1$$

$$DF(sun)=1+1+1$$

Jupiter is the fifth planet from the Sun and the largest planet in the Solar system. It is a gas giant.

Inverse Document Frequency (1)

- "**document exhaustivity**" is the number of index terms assigned to a document
- The higher of "**document exhaustivity**", the higher of probability that a document being relevant to a query
  - Think about an extreme case: a document contains all indexed terms (the vocabulary)
- Solutions
  - "**optimal exhaustivity**": balance the number of indexed terms for one document
  - Weight terms according to its **term specificity**

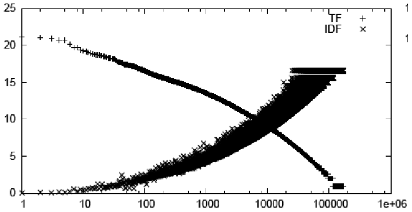
Inverse Document Frequency (2)

- **Specificity** is a property of the term semantics
  - "dog" is a more specific term than "domestic animal"
  - "husky" is a more specific term than "dog"
  - "tea" is a more specific term than "beverage"
- "term specificity" should be interpreted as a statistical rather than semantic property of the term
  - Because no "semantic tree" or "semantic network" is in practice use
  - "term specificity" is measured according to its statistical significance
  - Namely "term specificity" is measured by "inverse document frequency"

Inverse Document Frequency (3)

- Principle according to Zipf's law:
    - Higher weight is assigned to more specific term (less popular term)
    - Simple solution: $IDF(t_i) = \frac{1}{DF(t_i)}$
    - Normalize this term with size of dataset N:

$$IDF(t_i) = log\frac{N}{DF(t_i)} \tag{5}$$

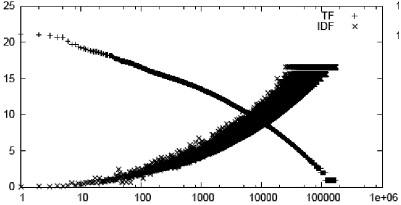


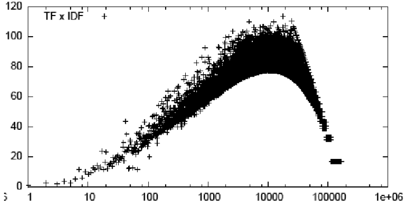
- Terms are ranked in descending order according to their DF
- IDF suppresses the weights of highly frequent terms

TF-IDF

- The best known term weighting schemes use the combination of IDF and term frequency
- Given f_{ij} is the term frequency of term t_i in document d_j,
- The term weight w_{ij} for t_i is defined as

$$w_{ij} = \begin{cases} (1 + log f_{ij}) \times log \frac{N}{DF_i}, & f_{ij} > 0 \\ 0 & otherwise \end{cases} \quad (6)$$





Distance measures: Euclidean distance (1)

- Given vector representation of query and document with TF-IDF weighting

$$
\begin{aligned}
q &= \{w_{1q}, w_{2q}, w_{3q} \ldots w_{iq} \ldots w_{nq}\} \\
d_j &= \{w_{1j}, w_{2j}, w_{3j} \ldots w_{ij} \ldots w_{nj}\}
\end{aligned}
$$

- Distance between q and d_j is usually measured by Euclidean distance (ℓ_2)

$$
d(q, d_j) = \sqrt{\sum_i (w_{iq} - w_{ij})^2} \tag{7}
$$

Distance measures: *Cosine* distance

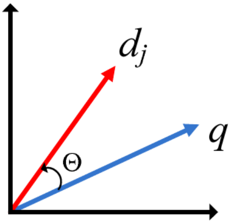
- Given vector representation of query and document with TF-IDF weighting

$$q = \{w_{1q}, w_{2q}, w_{3q} \dots w_{iq} \dots w_{nq}\}$$
$$d_j = \{w_{1j}, w_{2j}, w_{3j} \dots w_{ij} \dots w_{nj}\}$$

- *Cosine* distance is defined as

$$cos(q, d_j) = \frac{\sum_i w_{iq} \cdot w_{ij}}{\sqrt{\sum_i w_{iq}^2} \cdot \sqrt{\sum_i w_{ij}^2}} \tag{8}$$

Distance measures: *Cosine* distance (2)



$$cos(q, d_j) = \frac{\sum_i w_{iq} \cdot w_{ij}}{\sqrt{\sum_i w_{iq}^2} \cdot \sqrt{\sum_i w_{ij}^2}} \tag{9}$$

- Assignment: given q and d_j are ℓ_2-normalized, find the relation between Cosine distance and ℓ_2 distance

$$w_{iq} = \frac{w_{iq}}{\sqrt{\sum_i w_{iq}^2}} \quad , \quad w_{ij} = \frac{w_{ij}}{\sqrt{\sum_i w_{ij}^2}}$$

Brief summary over vector model

- Overcome most of the pitfalls of Boolean model
- We are ready to retrieve documents and rank them
  - Vector to vector comparison is not going to be efficient
- Vector model views that terms are independent from each other

Q & A

Thanks for your attention!