Textual information retrieval Models

Anne-Laure Ligozat

2018/2019 [1]

1. freely inspired by Benjamin Piwowarski and Hinrich Schütze's courses

IR systems

- Document indexing
- Query analysis
- Search model
- Evaluation

Plan

Why index ?

- Objective : find documents relevant for the user query
- From the query **words**
- Impossible to go through the entire collection
  - too many documents → prohibitive response time
  - operations between terms (not, near...) complex

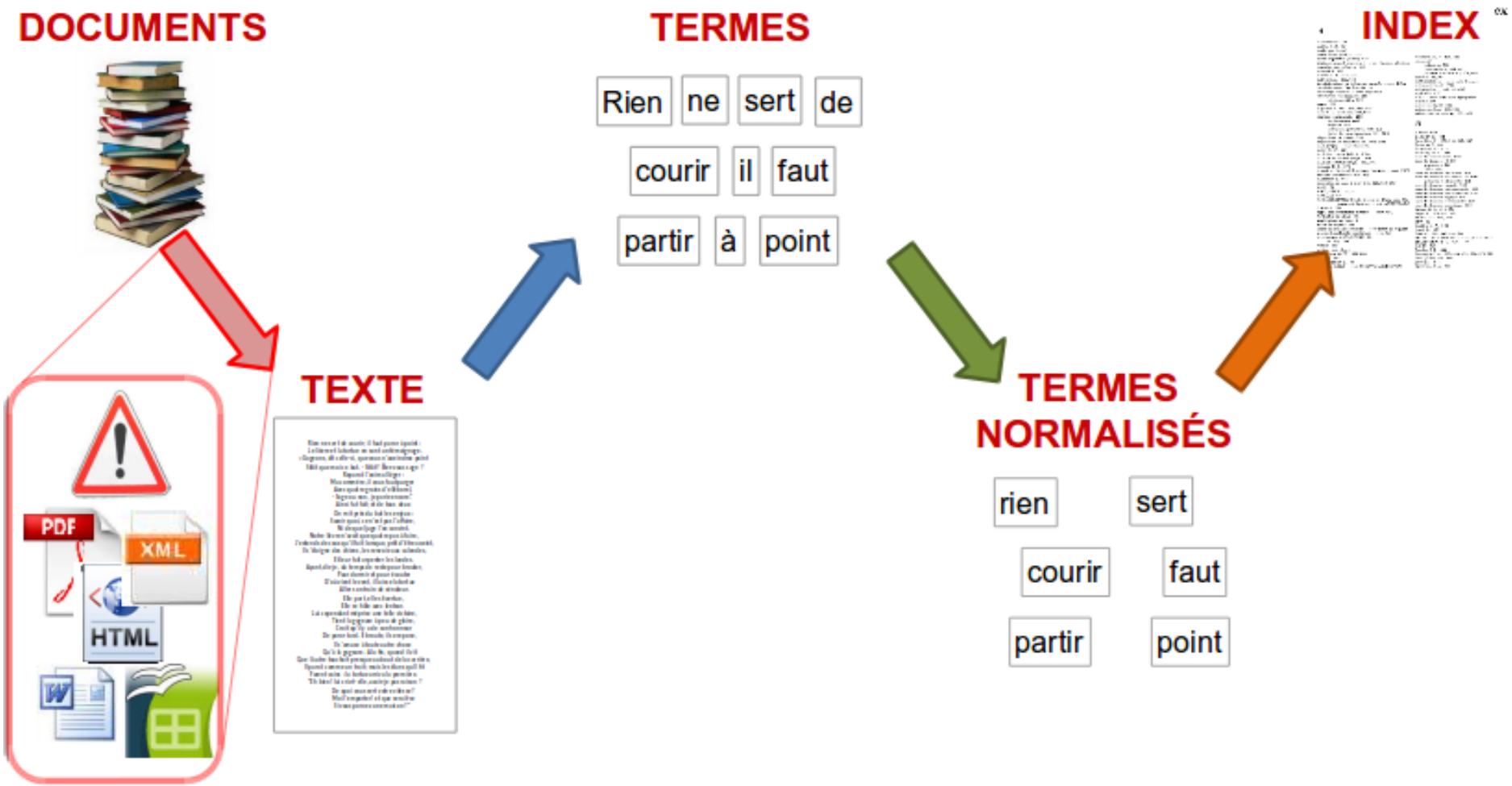
⇒ pre-processing = indexing

  - goal : "transform documents into substitutes capable to represent the documents content" (Salton et McGill, 1983)

Free vs controlled indexing

- free indexing : terms from documents
- controlled indexing : predefined terms
  - controlled vocabulary : avoids polysemy, synonymy, granularity problems

Documents


DOCUMENTS
TERMES
INDEX
Rien ne sert de
courir il faut
partir à point
TEXTE
PDF
XML
HTML
TERMES
NORMALISÉS
rien sert
courir faut
partir point

Formats

Formats

- HTML (diff : menus, tables, ads, appearance)
- raw text (structure ?)
- pdf (encoding, appearance)
- word (proprietay format, structure)
- excel (tables)
- openoffice (xml)

Taking the format into account

- detecting a document type is quite simple
- heuristics specific to each format to extract the text
- search engines raraly use the document structure

Language and encoding

Languages

- language identification = difficult problem
- multilingual information retrieval possible

Encoding

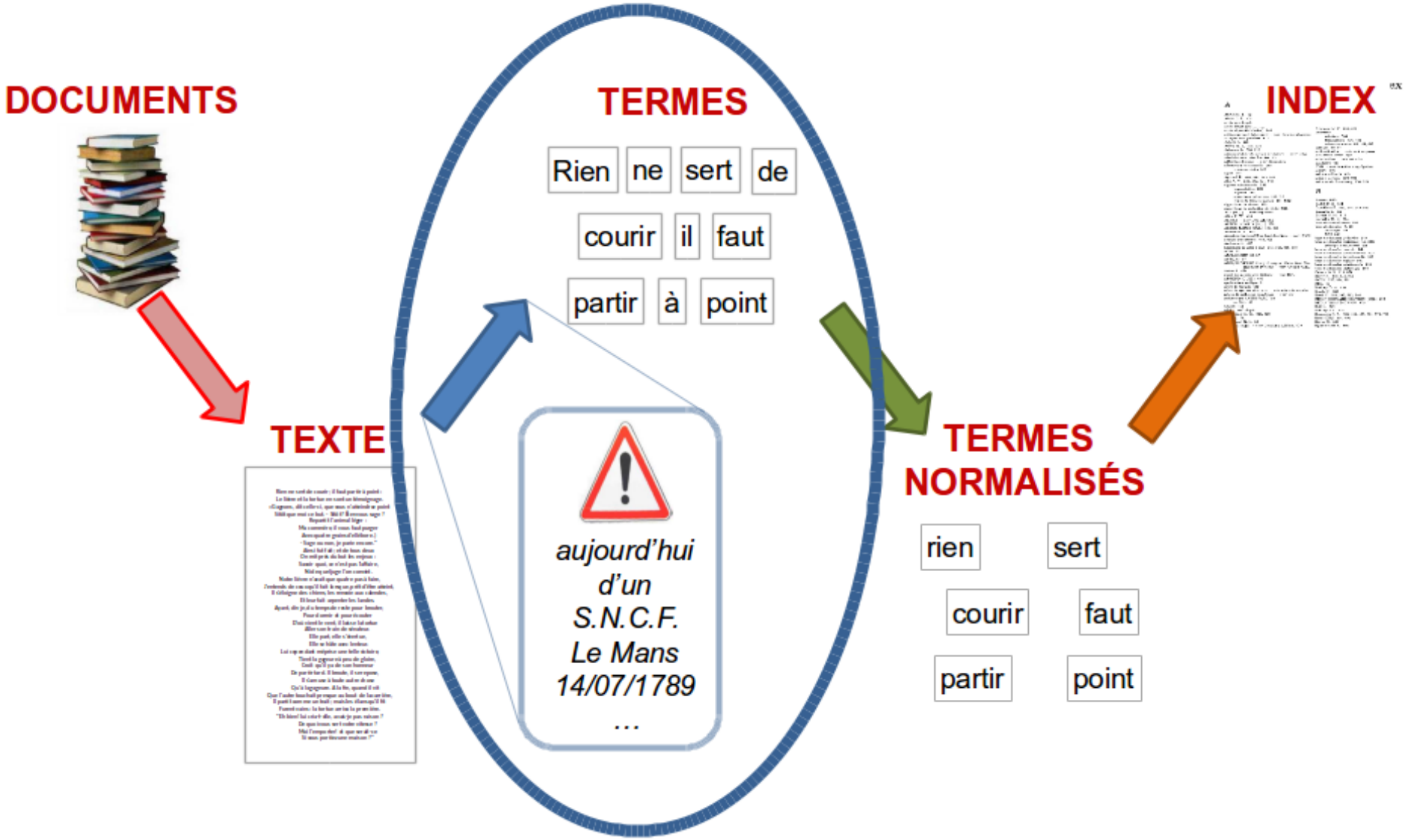
- errors in dealing with encoding → incorrect results

Document content

unit =

- file ?
- e-mail ?
  - with heading ?
  - with attachments ?
- set of files
  - website
  - document with multiple files
- ...

From text to terms



Tokenization

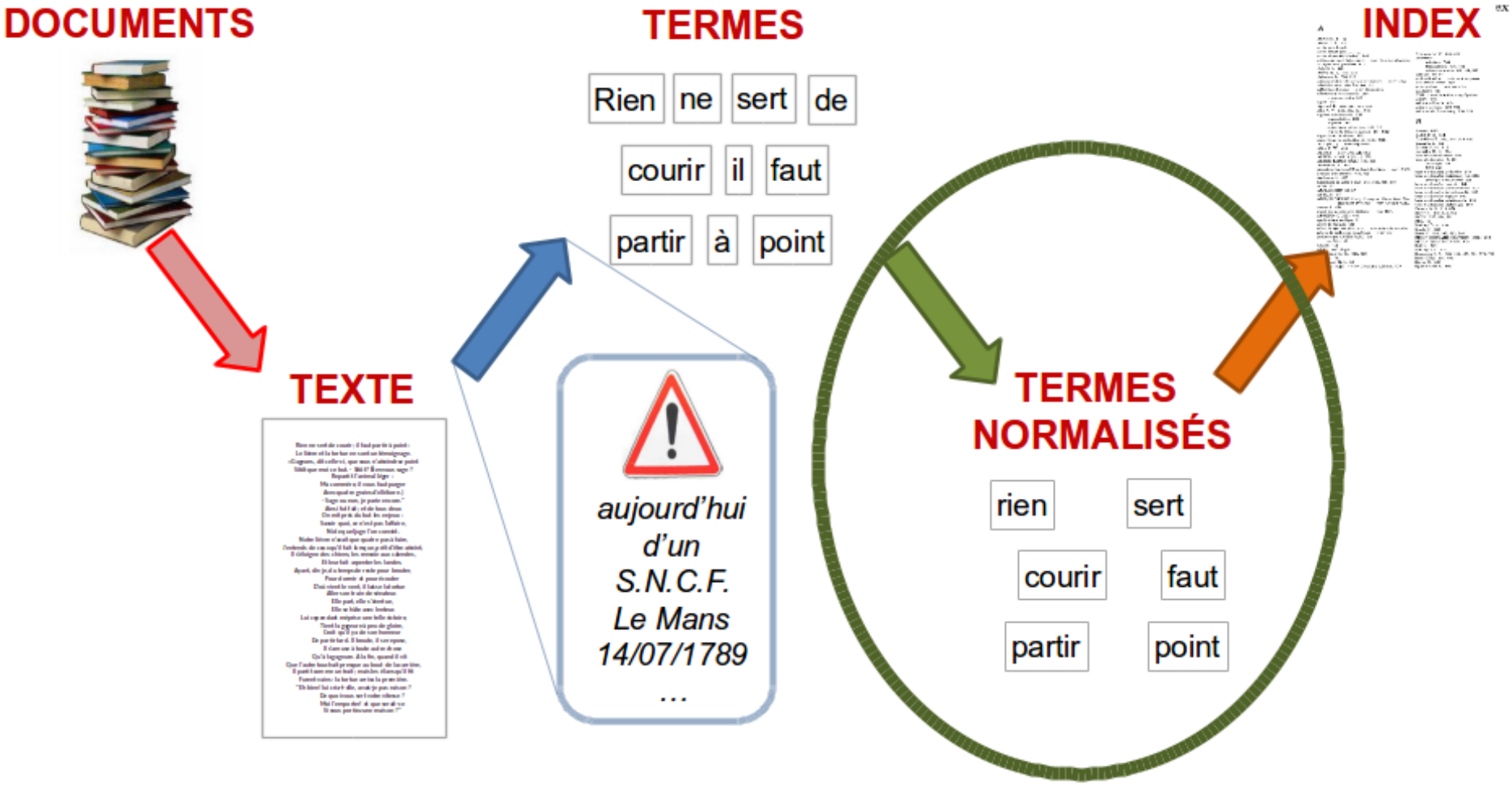
Tokenization = identification of elementary units

- **word** : substring as it appears in the text
- **term** (or type) : normalized word (case, morphology, spelling...)
  - set of terms = dictionary
- **token** : instance of a word or a term in a document

Difficulties of tokenization

- graphical variants of words with possible separators
  - États-Unis ou États Unis
- composed words in agglutinative languages
  - Lebensversicherungsgesellschaftsangestellter (employee from a lige insurance company)
- multiple alphabets in Japanese for example
- bidirectionnality of the reading sense in Arabic (numerals and letters)
- numbers : 555 3424, 24.09.2018
- ...

Normalization



Variant normalization (1/2)

- in documents and query
- variants that should be grouped
  - word variants including punctuation
    - U.S.A. and USA
    - morpho-syntaxe and morphosyntaxe
  - diacritic variants
    - in German, Tuebingen, Tübingen and Tubingen
    - in English, resume = résumé
  - proper names variants
    - Gorbatchov and Gorbatchev
- but
  - accents can be relevant
    - sur and sûr
    - pêche and péché
  - the case can be relevant
    - in German, mit and MIT
      (interaction between normalization and language detection)
    - in English, fed and Fed

Variant normalization (2/2)

- possibly asymetric
  - window → window, windows
  - windows → Windows, windows
  - Windows : non expansion
- + typos or spelling, OCR errors
- important criterion : how will the users write their query most often ?

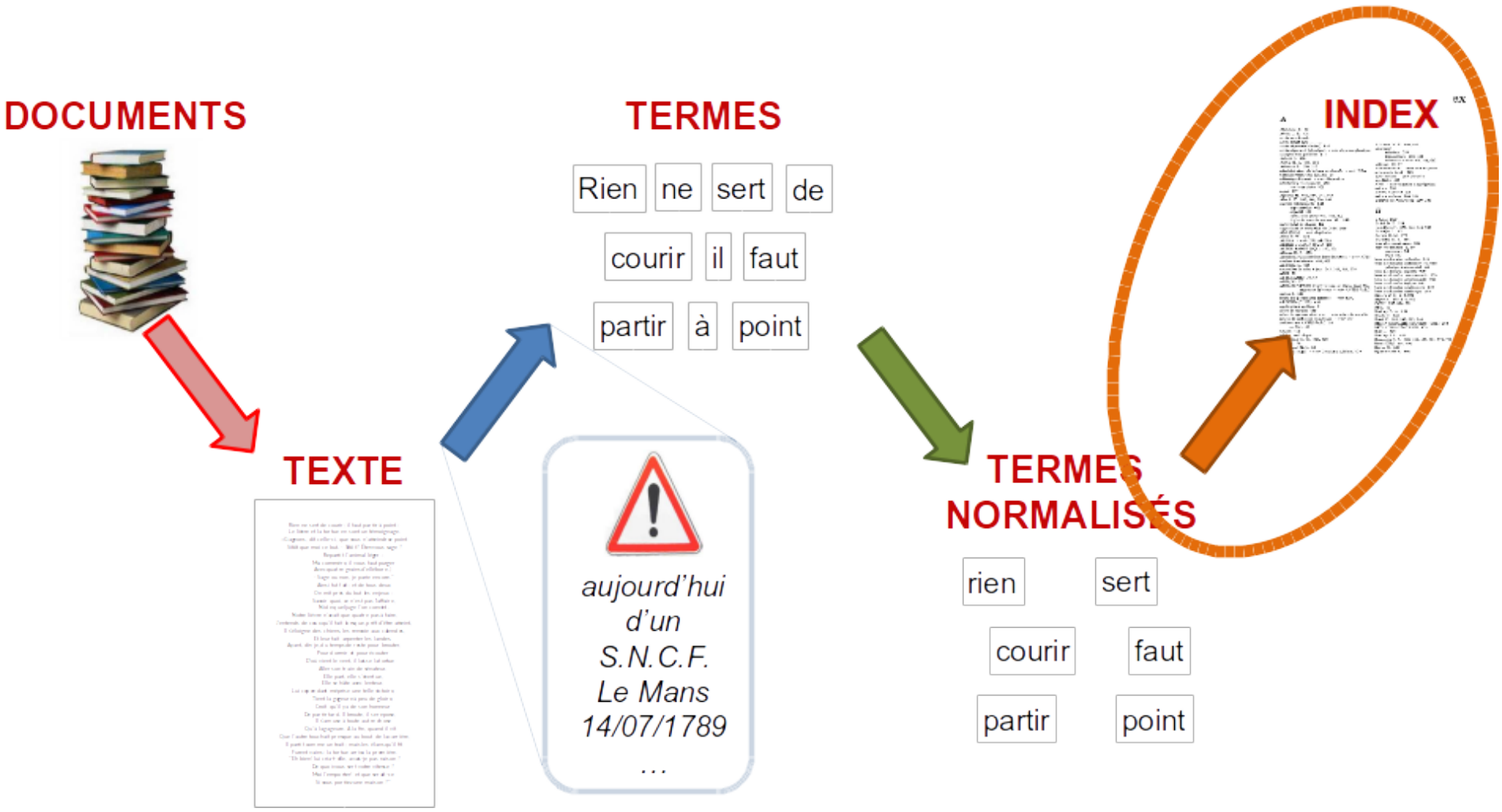
Morphological normalization

- use of analyses
  - lemmatization : chanteurs → chanteur, chantions → chanter
  - stemming : automate, automatique, automatisation → automat
    - in particular Porter algorithm, classical algorithm for English
    - stemming useful for certain queries, strongly worsens the results for others
  - POS tagging
- technics considered as mainly solved : low percentage of errors but difficult to reduce

Stop words

- word that do not add sense to the text
  - determiners : le, la
  - pronouns : je, nous
  - prepositions : sur, contre
- they are the most frequent in a language
  - the 30 most frequent words represent 30% of the word occurrences
  - delete them enables to gain much space in the index
- but
  - useful for multi-term queries : "pomme de terre", "les Chevaliers du Zodiaque"
  - sometimes bring sense in particular cases : "Let it be", "The Who", "être ou ne pas être"
  - compression actually enables to keep stopwords with little space

Index


DOCUMENTS
TERMES
INDEX
Rien ne sert de
courir il faut
partir à point
TEXTE
aujourd'hui
d'un
S.N.C.F.
Le Mans
14/07/1789
...
TERMES
NORMALISÉS
rien sert
courir faut
partir point

Incidence matrix

	Antoine & Cléopâtre	Jules César	La Tempête	Hamlet	Othello	Macbeth
Antoine	1	1	0	0	0	0
Brutus	1	1	0	1	0	0
César	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cléopâtre	1	0	0	0	0	0
pitié	1	0	1	1	1	1
pire	1	0	1	1	1	0

Incidence matrix

Brutus ET Cléopâtre ET PAS Calpurnia

	Antoine & Cléopâtre	Jules César	La Tempête	Hamlet	Othello	Macbeth
Antoine	1	1	0	0	0	0
Brutus	1	1	0	1	0	0
César	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cléopâtre	1	0	0	0	0	0
pitié	1	0	1	1	1	1
pire	1	0	1	1	1	0

Vecteurs d'incidence

¬Calpurnia	1	0	1	1	1	1

ET "bit à bit"

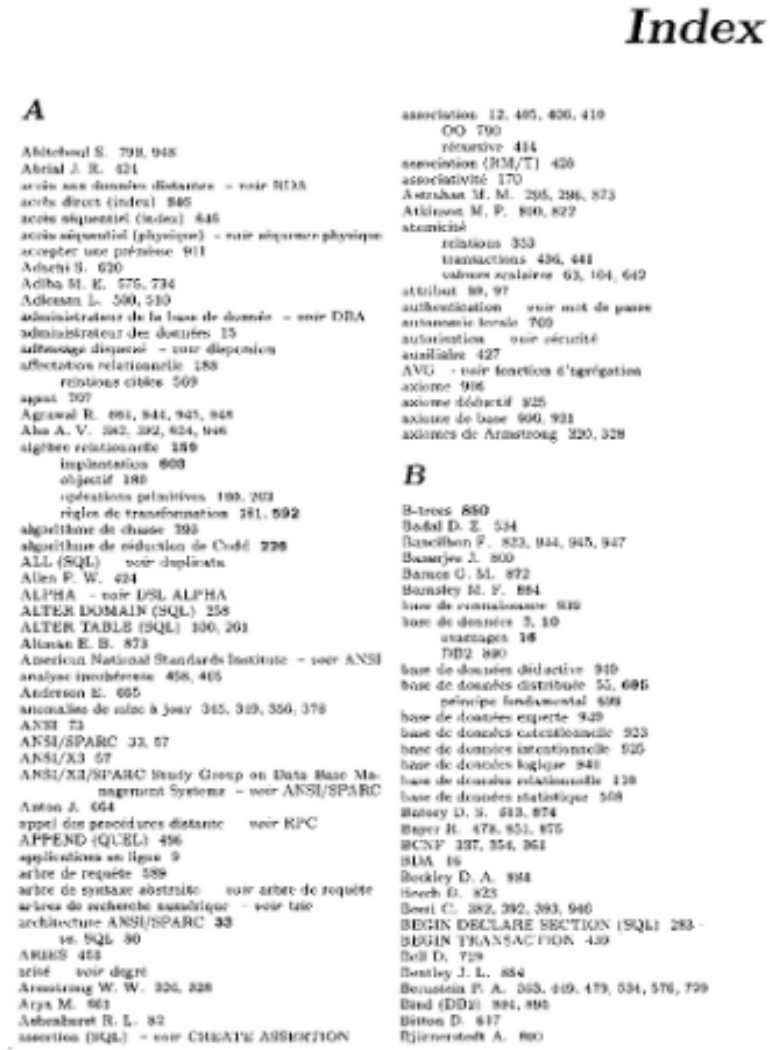
1	0	0	0	0	0

Incidence matrix

- impossible to use in practice
  - collection of a million documents
  - about 1000 words per document
  - total vocabulary of 500,000 distinct words

  → how many objects in the matrix ? how many 0s ? and 1s ?

Inverted index



Index

A

Abiteboul S. 798, 948
Abrial J. R. 424
accès aux données distantes – voir SDA
accès direct (index) 846
accès séquentiel (index) 846
accès séquentiel (physique) – voir séquence physique
accepter une prémisse 901
Adiba S. 620
Adiba M. E. 575, 734
Adleman L. 500, 510
administrateur de la base de donnée – voir DBA
administrateur des données 15
adressage disperse – voir dispersion
affectation relationnelle 188
    restions citées 509
agent 707
Agrawal R. 661, 844, 945, 948
Aho A. V. 382, 392, 624, 946
algèbre relationnelle 189
    implantation 603
    objectif 189
    opérations primitives 190, 203
    règles de transformation 181, 592
algorithme de chasse 393
algorithme de réduction de Codd 226
ALL (SQL) voir duplicata
Allen F. W. 424
ALPHA – voir DSL ALPHA
ALTER DOMAIN (SQL) 258
ALTER TABLE (SQL) 100, 261
Altman E. B. 873
American National Standards Institute – voir ANSI
analyse ascendante 458, 465
Anderson E. 665
anomalies de mise à jour 345, 349, 356, 378
ANSI 73
ANSI/SPARC 33, 57
ANSI/X3 57
ANSI/X3/SPARC Study Group on Data Base Management Systems – voir ANSI/SPARC
Anton J. 664
appel des procédures distante voir RPC
APPEND (QUEL) 456
applications en ligne 9
arbre de requête 588
arbre de syntaxe abstraite voir arbre de requête
arbres de recherche numérique – voir trie
architecture ANSI/SPARC 33
    vs. SQL 80
ARIES 458
arité voir degré
Armstrong W. W. 326, 328
Arya M. 661
Ashenhurst R. L. 82
assertion (SQL) – voir CREATE ASSERTION
association 12, 405, 406, 410
    OO 790
    récursive 414
association (RM/T) 420
associativité 170
Astrahan M. M. 295, 296, 873
Atkinson M. P. 800, 822
atomicité
    relations 353
    transactions 436, 441
    valeurs scalaires 63, 104, 642
attribut 89, 97
authentification voir mot de passe
autonomie locale 769
autorisation voir sécurité
auxiliaire 427
AVG – voir fonction d'agrégation
axiome 906
axiome déductif 325
axiome de base 906, 931
axiomes de Armstrong 320, 328

B

B-trees 850
Badal D. Z. 534
Bancilhon F. 823, 944, 945, 947
Banerjee J. 800
Barnes G. M. 872
Barnsley M. F. 884
base de connaissance 930
base de données 3, 10
    avantages 16
    DB2 880
base de données déductive 910
base de données distribuée 55, 695
    principe fondamental 699
base de données experte 929
base de données extensionnelle 923
base de données intentionnelle 925
base de données logique 940
base de données relationnelle 139
base de données statistique 509
Batory D. S. 613, 874
Bayer R. 478, 851, 875
BCNF 337, 354, 361
BDA 16
Beckley D. A. 884
Beech D. 825
Beeri C. 382, 392, 393, 946
BEGIN DECLARE SECTION (SQL) 283
BEGIN TRANSACTION 439
Bell D. 729
Bentley J. L. 884
Bernstein P. A. 365, 449, 479, 534, 576, 729
Bind (DB2) 894, 895
Bitton D. 617
Bjorneroteft A. 880

Inverted index

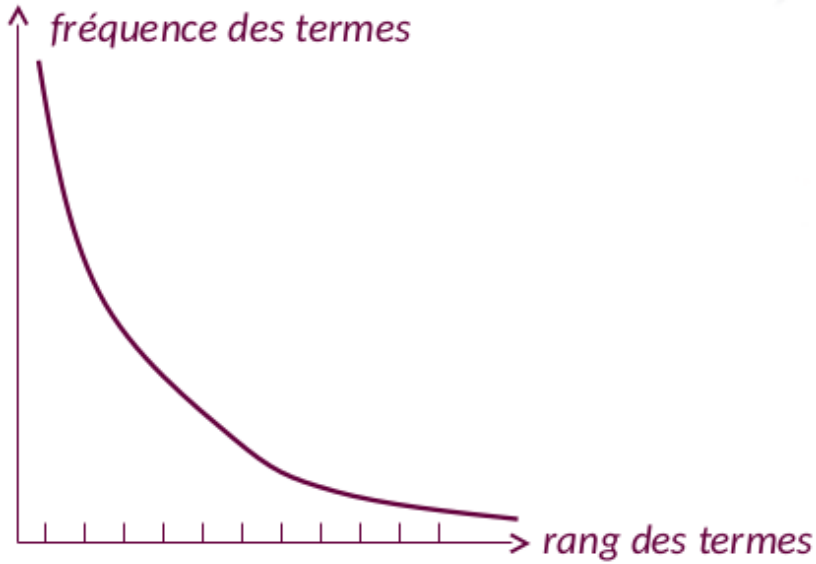
- classical notion of index
- associates indexes to documents that contain them (unique identifier)
  - a $\rightarrow$ d1, d2, d3, d4, d5...
  - à $\rightarrow$ d1, d2, d3, d4, d5...
  - abaissa $\rightarrow$ d3, d4...
  - abaissable $\rightarrow$ d5
  - abandon $\rightarrow$ d1, d5
  - abandonna $\rightarrow$ d2
  - ...

Vocabulary size

- The vocabulary increases with the colection
- Heaps law : M $= kT^b$ with
  - M vocabulary size
  - T number of tokens in the collection
  - b and k constants (typically b=0,5 and k = 30 to 100)
  - empirical law
- much worse for the Web !

Term frequency

- few frequent words and many rare words
- Zipf law : the n^th most frequent word has a frequency (= number of occurrences) proportional to $1/n$



tfidf

- in a query like in a document, all terms do not have the same importance
- intuition 1 : the more occurrences of a word a document contains, the more it is about this term $\rightarrow$ $\text{tf}_{t,d}$ = number of occurrences of term t in document d
- intuition 2 : very frequent words in all documents are less important (less discriminating) $\rightarrow$ df_t = number of documents that contain term t
- weight of a term $tf.idf_t, d = tf_{t,d} x log_{10} \frac{N}{df_t}$ (N = number of documents)

Weighted matrix

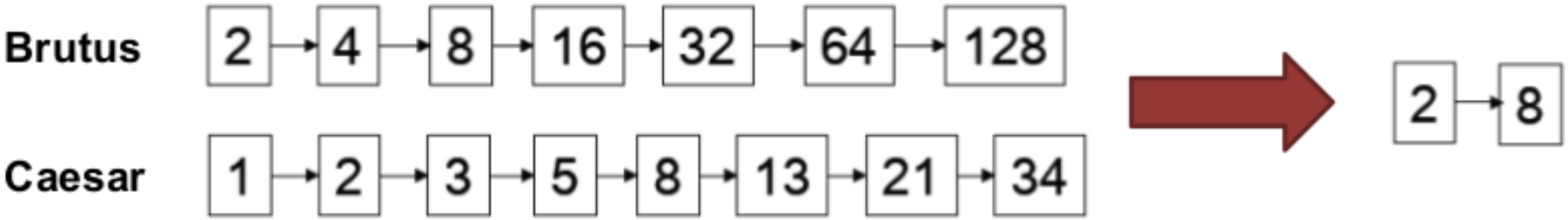
	Antoine & Cléopâtre	Jules César	La Tempête	Hamlet	Othello	Macbeth
Antoine	13,1	11,4	0	0	0	0
Brutus	3,0	8,3	0	1	0	0
César	2,3	2,3	0	0,5	0,3	0,3
Calpurnia	0	11,2	0	0	0	0
Cléopâtre	17,7	0	0	0	0	0
pitié	0,5	0	0,7	0,9	0,9	0,3
pire	1,2	0	0,6	0,6	0,6	0

each document is a vector in $\mathbb{R}^{|V|}$

Finding the documents

Brutus AND Caesar

- On **recherche** « **Brutus** » dans le dictionnaire
  → On récupère la liste de documents
- On **recherche** « **Caesar** » dans le dictionnaire
  → On récupère la liste de documents
- On **fusionne** les deux listes



Queries with phrases

- If a user formulates a query such as "Paris Saclay", a document containing "Le maire de Paris s'est arrêté dans un restaurant de Saclay aujourd'hui" is not likely to be relevant
- concept of phrase easy to understand for users
- a large part of web queries
- simple index insufficient
- two solutions :
  - n-grams index
  - positional index

N-gram

- n-gram = subsequence of n elements extracted from a given sequence
- here, word n-grams
  - unigrams : all words
  - bigrams : sequences of 2 words
- différent from a linguistic phrase

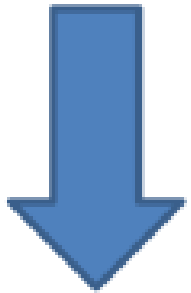
Bigrams index

- index, in addition to simple words, the text bigrams
- 1 bigram = 1 term from dictionary
- actually rarely used
  - query n-grams difficult to determine (Stanford University Palo Alto, Université Paris-Saclay Orsay)
  - index vocabulary very large

Positional index

- Idea : in the lists of documents in the index, add the position of each term occurrence in the document

terme	fréquence	→	D1	D3	D4



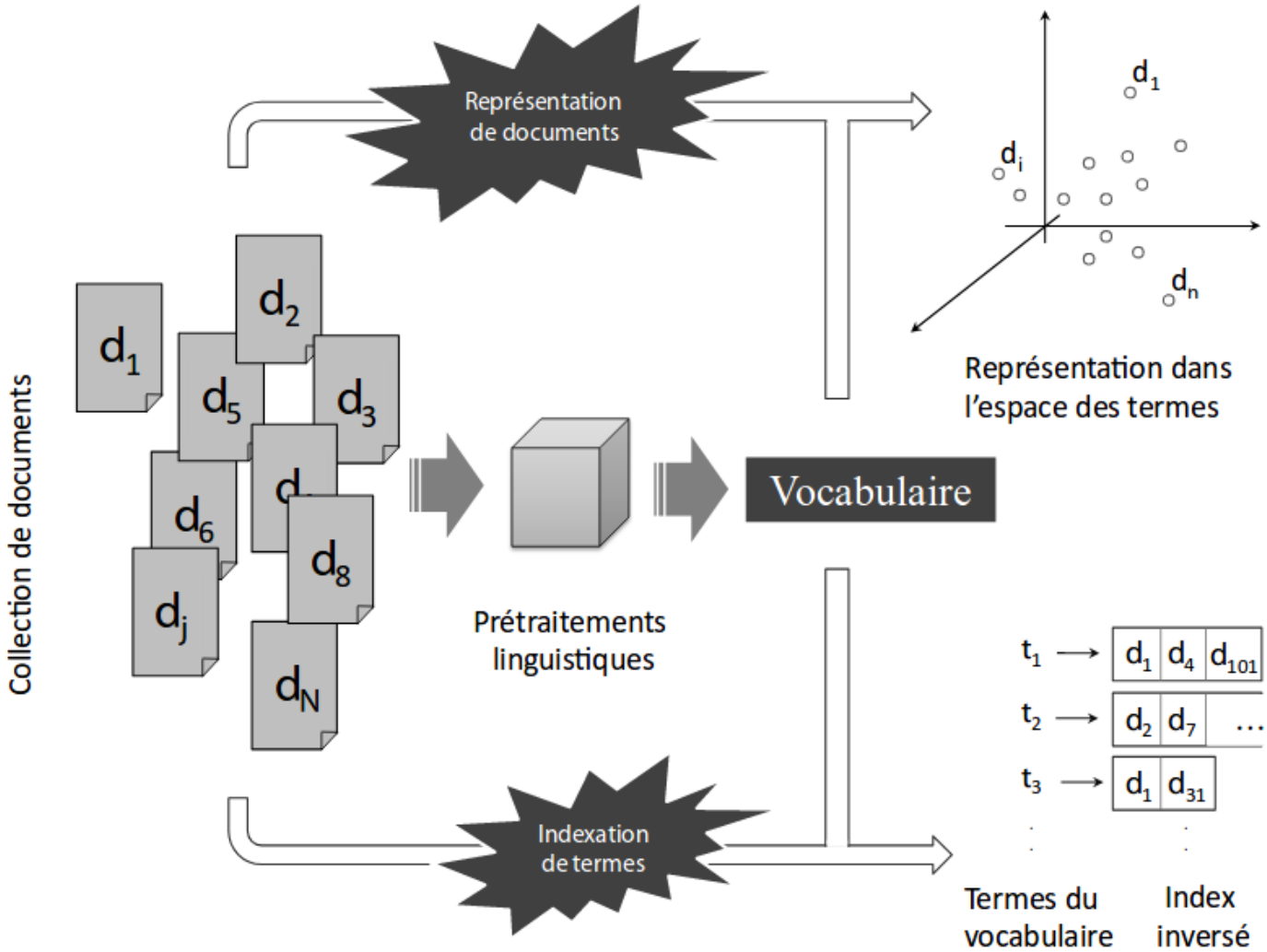
terme	fréquence	→	D1 : pos1, pos2, pos3
			D3 : pos1, pos2
			D4 : pos1, pos2, pos3

Traversing a positional index

- "Université Paris Saclay"
  - extraction of all dictionary entries
  - recursive use of the fusion algorithm, for documents then for positions
  - use of an incremental comparison instead of a strict equality

université	1252	→	D2 : 546
			D6 : 34, **87**, 145, 243
			D7 : 44, 87, 34
			...
paris	45	→	D2 : 547
			D6 : **88**, 543
			...
saclay	15345	→	D2 : 54, 90
			D6 : **89**
			D4 : 43

Plan



Recherche d'information, Applications, modèles et algorithmes de Massih-Reza Amini et Éric Gaussier

Search models : the three currents

- models based on set theory
  - boolean model
- algebric models
  - vectorial model
- probabilistic models
  - notion of relevance

from the beginning of IR (60s, 70s)

Boolean model

- first and simplest model
- based on set theory and Boolean algebra
- query tems either present or absent : binary weighting of terms, 0 or 1
- document soit pertinent soit non pertinent : pertinence binaire (modèle exact)
- query expressed with logical operators : AND, OR, NOT
  - (cyclisme OR natation) AND NOT dopage
  - document relevant iff its content respects the logical formula

Boolean model : example

Requête Q : `(cyclisme OR natation) AND NOT dopage`

Le document contient					Pertinence du document
cyclisme	natation	**cyclisme OR natation**	dopage	**NOT dopage**	
0	0	**0**	0	**1**	**0**
0	0	**0**	1	**0**	**0**
0	1	**1**	0	**1**	**1**
0	1	**1**	1	**0**	**0**
1	0	**1**	0	**1**	**1**
1	0	**1**	1	**0**	**0**
1	1	**1**	0	**1**	**1**
1	1	**1**	1	**0**	**0**

Boolean model : pros and cons

Pros

- precise : document contains terms or non
- understandable
- still used in many tools, such as electronic mail
- adapted for specialists when constraint vocabulary preferred (law)

Cons

- difficult to express long queries with boolean form
- binary criterion not very efficient
  - often too many of too few results
  - term weighting improves results (cf. exended boolean model)
- impossible ro rank the results
  - all returned documents are at the same level
  - users prefer a ranking when the list is long

Ranked lists of results

Why rank results ?

- most users
  - do not know how to write boolean queries
  - do not want to parse too many results (possibly millions of them)
- prefer ranked lists
  - from most useful to the user (relevant) to less useful
  - the number of results is no more a problem
  - the user parses as many as they want
- but requires an effective ranking algorithm
- statistical model
  - quantitative aspect of terms and documents
  - similarity measure between query and document

Ranked models

Ranking

- the number of results is not a problem any more : first 10
- if the ranking algorithm works correctly

Principle

- attribure a score to each query-document pair
  - according to the document relevance to the query
  - generally presence of query terms in the document
- rank documents by descending score

Vector space model

Vector space model

- Similarity measure : close representations ⇒ high probability that same information
- Documents and queries represented by vectors in a euclidian space with n dimensions (n : number of terms)
  - terms = axes
  - docs = vectors (sparse)
- Document relevance = degree of similarity between vectors of query and document
- Documents ranked from most similar to the query to less similar

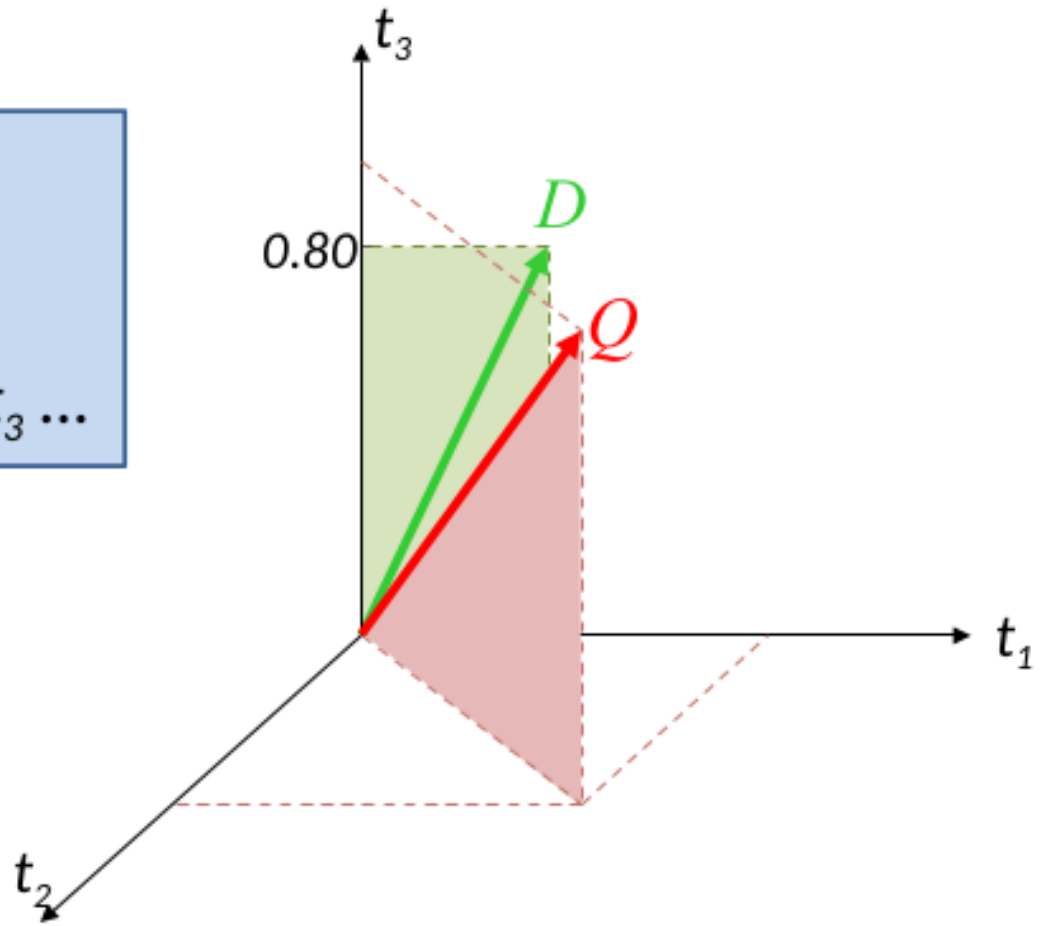
Vector space model

Requête Q : $t_1\ t_2\ t_3$

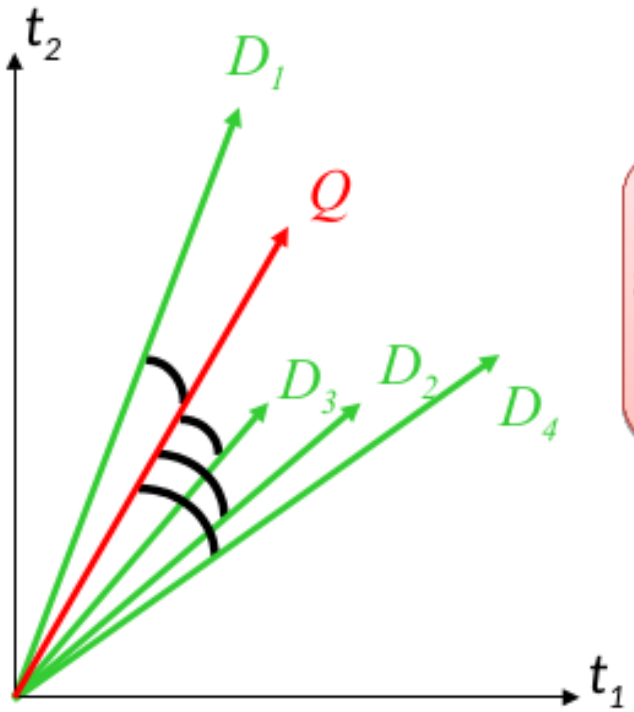
Document D : ... t_1 ... t_3 ...

Poids $w_{D,t1} = 0.45$

Poids $w_{D,t3} = 0.80$



Similarity measure



Cosinus

$$\text{sim}(\vec{Q},\vec{D}) = \frac{\vec{Q}\cdot\vec{D}}{|\vec{Q}|\times|\vec{D}|} = \frac{\sum_{i=1}^{n} w_{i,Q}\times w_{i,D}}{\sqrt{\sum {w^2}_{i,Q}}\times\sqrt{\sum {w^2}_{i,D}}}$$

(Le produit scalaire avec normalisation de la longueur des vecteurs)

Vector space models : pros and cons

Pros

- simple query language : keywords list
- performance better than boolean thanks to term weighting
- partial relevance of documents possible
- ranking of documents possible

Cons

- terms considered as independent
- query language less expressive
- understanding harder

Probabilistic model

Problem modelling

- estimation of the relevance probability of a document for a query
- binary notion of relevance :
  - $R_{d,q} = 1$ if d relevant for q
  - $R_{d,q} = 0$otherwise
- documents ranked by descending relevance probability
- relevance of each document supposed to be independent
- dyssymetry between query and document ($\neq$ vector space)

Probabilistic model : conclusion

- leading model :
  - Okapi BM25
    - non binary probabilistic model (term frequency) with document length normalization
    - robust and used a lot
- other probabilistic models
  - bayesian networks
  - language model
    - document = generative model that generates the query
- conclusion
  - problem of initial probabilities
  - independent terms
  - results close to vector space model

Learning to rank

Basic principle

- features that influence the relevance
  - query : length, idf means...
  - document : PageRank, spam probability, document date...
  - query + document : cosine similarity between query and document, minimal window in which query terms appear, zones...
- binary classification : relevant 1, not relevant 0
  - but how to learn regression score
- (document) pair learning
  - but errors do not have the same importance
  - sensitive to the number of relevant documents per query
  - first returned documents more important
- apprentissage de listes : optimiser MAP directement par exemple
  - difficult ML problem

Other models

- generalized vector space model
  - represents dependencies between terms
  - theoretically interesting, but effectiveness not proved
- Latent Semantic Indexing
  - proposes to study "concepts" instead of terms (idead of a text)
  - links documents between them and to query
  - enables to return documents containing no query word
  - less dimensions
  - better recall, lower precision

Plan

What is a good search engine ?

Criteria

- main criterion : user satisfaction ?
- fast
  - fast query analysis
  - fast index search
  - fast result ranking
- complete and up-to-date
  - all (or many) documents from the collection are used
  - new documents quickly added

  $\Rightarrow$ fast index building

  $\Rightarrow$ (on the Web) permanent, effective and fast discovery of the new documents
- most important : relevant

How to measure relevance ?

- Web search engine
  - the user clicks on some links and not others
  - the user comes back to the search engine
  - the user manages to perform a certain task
- e-commerce website
  - the user buys
  - the user buys quickly
  - a large proportion of visitors buy
- firm website
  - the user gains productivity
  - secured access
  - ...

What is a good evaluation ?

- evaluating a system aims at knowing :
  - if it performs the expected task
  - if it is better than the competition
  - how to improve it
- that means that the evaluation should be
  - reproducible
    - to evaluate several systems the same way
    - to assess the progress made
  - understandable
    - to identify the possible progress zones
  - fast
    - to evaluate each modification independently
  - objective

How to make relevance objective ?

- user need transformed into a query $\rightarrow$ 1st information loss
  - information need : je voudrais savoir si boire du vin rouge réduit le risque de problèmes de coeur
  - query : vin rouge problèmes coeur
  - doc : le coeur de son discours concernait le problème de l'industrie du vin qui peine à reconnaître le rôle de la consommation de vin rouge dans les accidents de voiture

  $\rightarrow$ doc relevant for the query but not for the need
- yet, relevance of the results compared to the information need
- relevance not binary : very relevant, not at all, a little, why not...
- in order to make relevance objective, simplified definition :
  - documents treated independently from each other
  - relevance transformed into a binary notion
- and use of test collections

Standard methodology

- document collection
  - representative of real documents
- set of information needs/queries
  - also representative
- relevance score for each document and each query
  - human judgements

standard benchmarks : TREC

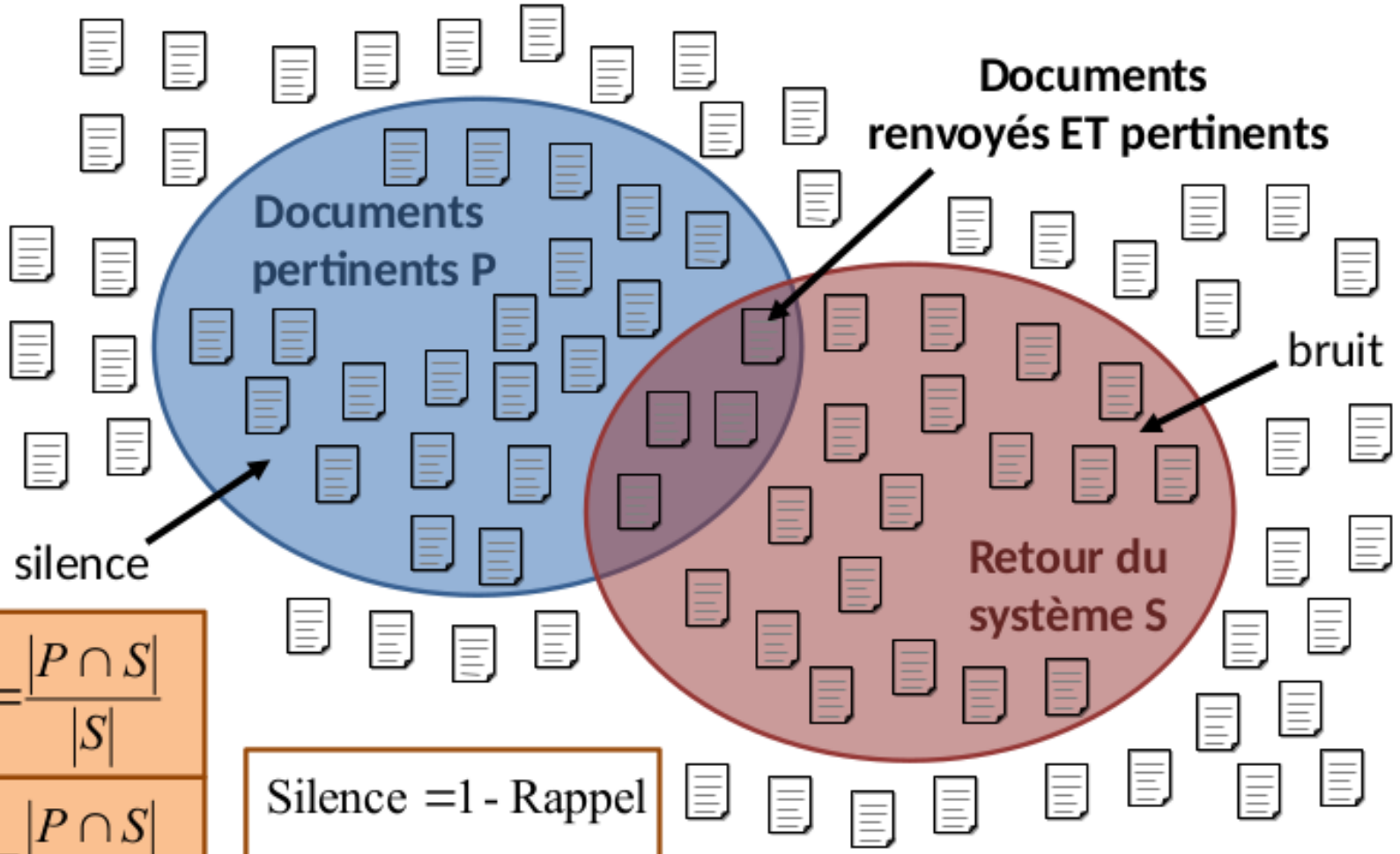
- Ad Hoc in particular (1992 to 1999)
- scores for the first k documents returned by systems

Test collections

Test collections make the experiments reproducible

- A protocol is defined
- A significative number of examples is manually judged
  - gold standard
  - A part can also serve as a development and/or training set
- An inter-annotator agreement is computed
  - To validate the objectivity
- System results are compared to expected results
- Imperfect bur precise metrics are defined

Evaluation : precision and recall



$$\text{Précision} = \frac{|P \cap S|}{|S|}$$

$$\text{Rappel} = \frac{|P \cap S|}{|P|}$$

Silence =1 - Rappel

Bruit =1 - Précision

Complementarity of precision and recall

Why not just precision ?

- precision = capacity of a system to return MOSTLY relevant documents
- Returning a single relevant document $\Rightarrow$ 100% precision
- not compatible with user satisfaction !

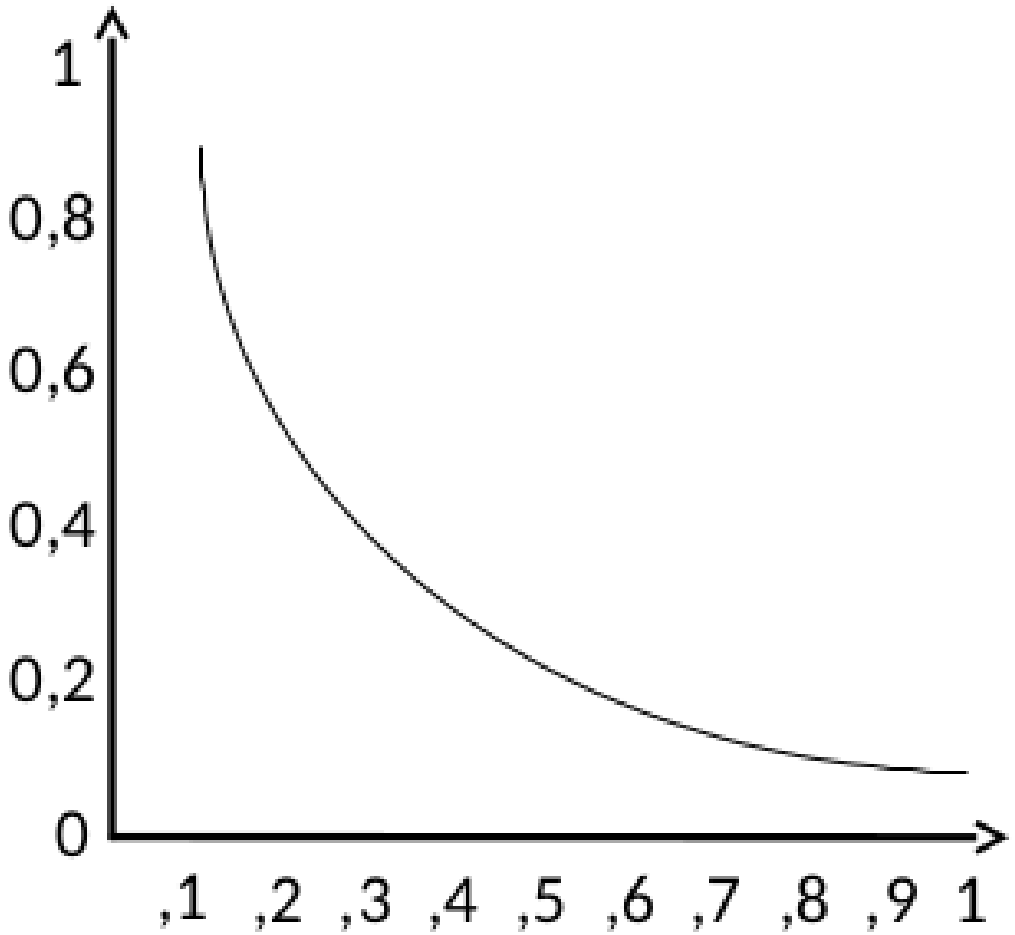
Why not just recall ?

- recall = capacity of a system to return ALL relevant documents
- Returning all the collection $\Rightarrow$ 100% recall
- not compatible with user satisfaction !

Recall/precision curve

- recall improves with number of answers
- precision decreases

recall/precision curve used to characterize IR systems



F-measure

to obtain a unique value, F-measure = harmonic mean

$F = \frac{1}{\alpha \frac{1}{p} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)xPxR}{\beta^2 P + R}$ with $\alpha = \frac{1}{\beta^2+1}$

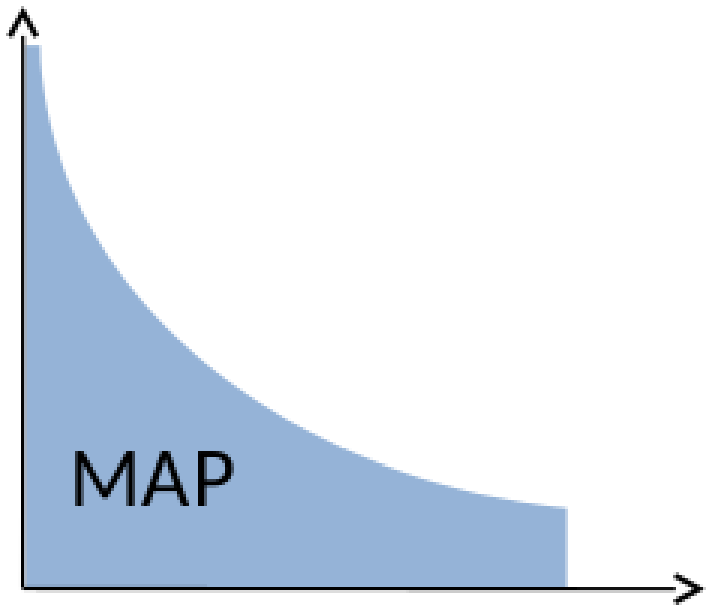
$\beta < 1$ favors precision, $\beta > 1$ recall

to give as much importance to precision and recall, $\beta = 1$

$F = \frac{2PR}{P+R}$

Other metrics

- MAP (Mean Average Precision) : area under the R/P curve
- to take into account result ranking :
  - P@5, P@10 : precision for first documents retrieved ; favors high precision
  - P@100
  - R/P curve for k variable
- error rate = (false positives + false negatives) / relevant
- and others...



Présentation des résultats

- the user must be able to identify the possibly relevant documents with their description $\rightarrow$ title, url, metadata
- often abstract too
  - static (independent of query)
  - dynamic : "snippets" from document that contains the query terms

References

- Recherche d'information, Applications, modèles et algorithmes de Massih-Reza Amini et Éric Gaussier (2e édition en 2017)
- Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (2008)