Information Retrieval

Lecture 2: Retrieval models

Computer Science Tripos Part II

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

Today

- Definition of the information retrieval problem
- Query languages and retrieval models
  - Boolean model
  - Vector space model
- Logical model of a document/a term
  - Term weighting
  - Term stemming

Information Retrieval: the task

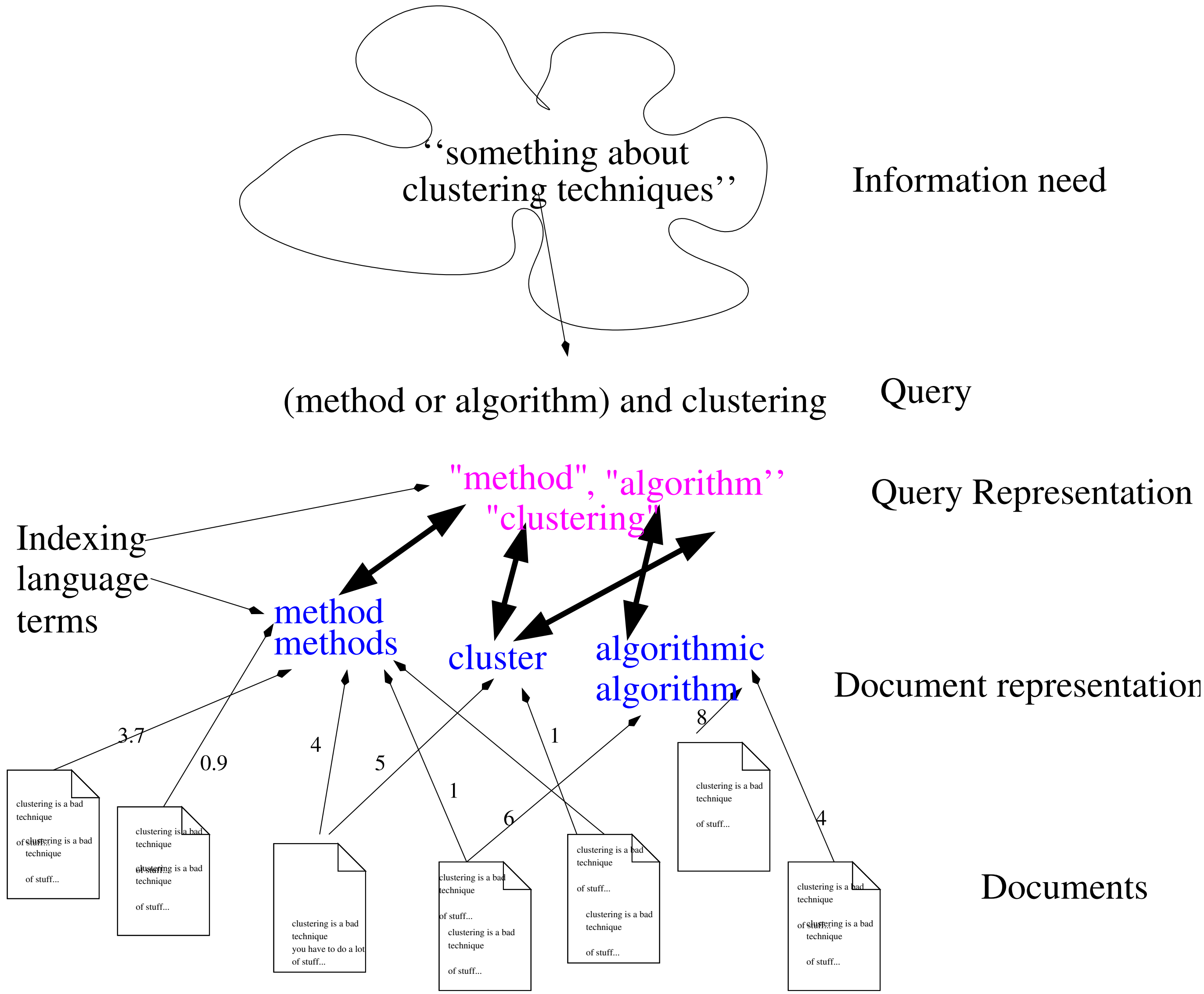
Problem: given a query, find documents that are “relevant” to the query

- Given: a large, static document collection
- Given: an information need (reformulated as a keyword-based query)
- Task: find all and only documents that are relevant to this query

Issues in IR:

- How can I formulate the query? (Query type, query constructs)
- How does the system find the best-matching document? (Retrieval model)
- How are the results presented to me (unsorted list, ranked list, clusters)?

Query and document representation



Indexing

- Indexing: the task of finding terms that describe documents well
- Manual indexing by cataloguers, using fixed vocabularies (“thesauri”)
  - labour and training intensive
- Automatic indexing
  - Term manipulation (certain words count as the same term)
  - Term weighting (certain terms are more important than others)
  - Index terms can only be those words or phrases that occur in the text

Fixed indexing languages/vocabularies (“thesauri”)

- Large vocabularies (several thousand items)
- Examples: ACM – subfields of CS; Library of Congress Subject Headings
- Problems:
  - High effort in training in order to achieve consistency
  - Subject matters emerge → schemes change constantly
- Advantages:
  - High precision searches
  - Works well for valuable, closed collections like books in a library

Examples, fixed indexing languages

Medical Subject Headings (MeSH)

...	
Eye Diseases	C11
Asthenopia	C11.93
Conjunctival Diseases	C11.187
Conjunctival Neoplasms	C11.187.169
Conjunctivitis	C11.187.183
Conjunctivitis, Allergic	C11.187.183.200
Conjunctivitis, Bacterial	C11.187.183.220
Conjunctivitis, Inclusion	C11.187.183.220.250
Ophthalmia Neonatorum	C11.187.183.220.538
Trachoma	C11.187.183.220.889
Conjunctivitis, Viral	C11.187.183.240
Conjunctivitis, Acute Hemorrhagic	C11.187.183.240.216
Keratoconjunctivitis	C11.187.183.394
Keratoconjunctivitis, Infectious	C11.187.183.394.520
Keratoconjunctivitis Sicca	C11.187.183.394.550
Reiter's Disease	C11.187.183.749
Pterygium	C11.187.781
Xerophthalmia	C11.187.810
...	

ACM Computing Classification System (1998)

B	Hardware
B.3	Memory structures
B.3.0	General
B.3.1	Semiconductor Memories (NEW) (was B.7.1)
	Dynamic memory (DRAM) (NEW)
	Read-only memory (ROM) (NEW)
	Static memory (SRAM) (NEW)
B.3.2	Design Styles (was D.4.2)
	Associative memories
	Cache memories
	Interleaved memories
	Mass storage (e.g., magnetic, optical, RAID)
	Primary memory
	Sequential-access memory
	Shared memory
	Virtual memory
B.3.3	Performance Analysis and Design Aids
	Formal models
	Simulation
	Worst-case analysis
B.3.4	Reliability, Testing, and Fault-Tolerance
	Diagnostics
	Error-checking
	Redundant design
	Test generation
...	

Free indexing languages

- No predefined set of index terms
- Instead: use natural language as indexing language
- Mappings words → meanings is not 1:1
  - Synonymy (n words : 1 meaning) — sofa – couch
  - Polysemy (1 word : n meanings) — bank – bank
- Do the terms get manipulated?
  - De-capitalised? — Turkey – turkey
  - Stemmed? — advice – advised
  - Stemmed and POS-tagged? — can – can
- Use important phrases, instead of single words
  cheque book (rather than cheque and book)

Implementation of indexes: inverted files

Inverted files

Doc 1
Except Russia and Mexico no country had had the decency to come to the rescue of the government.

Doc 2
It was a dark and stormy night in the country manor. The time was past midnight.

Term	Doc no	Freq	Offset
a	2	1	2
and	1	1	2
and	2	1	4
come	1	1	11
country	1	1	5
country	2	1	9
dark	2	1	3
decency	1	1	9
except	1	1	0
government	1	1	17
had	1	2	6,7
in	2	1	7
it	2	1	0
manor	2	1	10
mexico	1	1	3
midnight	2	1	17
night	2	1	6
no	1	1	4
of	1	1	15
past	2	1	15
rescue	1	1	14
russia	1	1	1
stormy	2	1	5
the	1	2	8,13
the	2	2	8,12
time	2	1	14
to	1	2	10,12
was	2	2	16

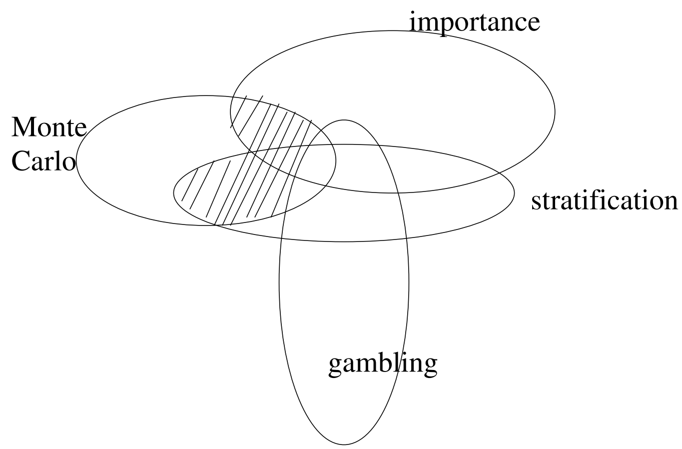
Information kept for each term:

- Document ID where this term occurs
- Frequency of occurrence of this term in each document
- Possibly: Offset of this term in document

Information Retrieval systems: Methods

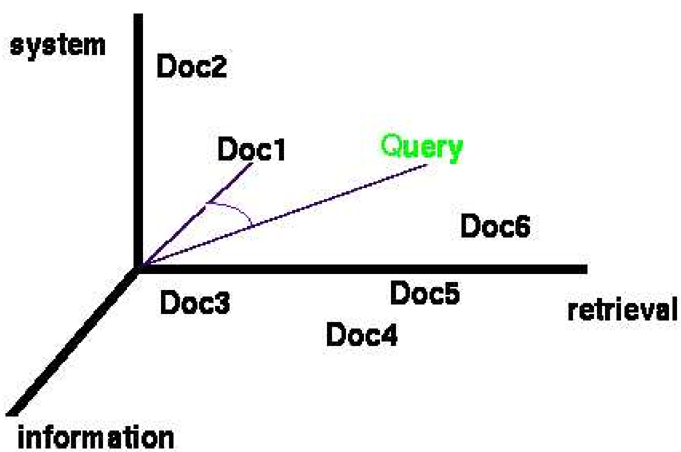
- Boolean search
  - Binary decision: Document is relevant or not (no ranking)
  - Presence of term is necessary and sufficient for match
  - Boolean operators are set operations (AND, OR)
- Ranked algorithms
  - Ranking takes frequency of terms in document into account
  - Not all search terms necessarily present in document
  - Incarnations:
    * The vector space model (SMART, Salton et al, 1971)
    * The probabilistic model (OKAPI, Robertson/Spärck Jones, 1976)
    * Web search engines

`Monte Carlo AND (importance OR stratification) BUT gambling`



- Set theoretic interpretation of connectors AND OR BUT
- Often in use for bibliographic search engines (library)
- Problem 1: Expert knowledge necessary to create high-precision queries
- Problem 2: Binary relevance definition → unranked result lists (frustrating, time consuming)

- A document is represented as a point in high-dimensional vector space
- Query is also represented in vector space
- Select document(s) with highest document–query similarity
- Document–query similarity is model for relevance → ranking



3-dimensional term vector space:

- Dimension 1: “information”
- Dimension 2: “retrieval”
- Dimension 3: “system”

Documents and queries in term feature space

	Doc_1	Doc_2	Doc_3	...	Doc_n		Q
$term_1$	14	6	1	...	0	$\leftrightarrow$	0
$term_2$	0	1	3	...	1	$\leftrightarrow$	1
$term_3$	0	1	0	...	2	$\leftrightarrow$	0
	...	...	...	...	...	$\leftrightarrow$	...
$term_N$	4	7	0	...	5	$\leftrightarrow$	1

Decisions to take:

1. Choose dimensionality of vector: what counts as a term?
2. Choose weights for each term/document mapping (cell)
   - presence or absence (binary)
   - term frequency in document
   - more complicated weight, eg. TF*IDF (cf. later in lecture)
3. Choose a proximity measure

A proximity measure can be defined either by similarity or dissimilarity. Proximity measures are

- Symmetric ($\forall i, j : d(j, i) = d(i, j)$)
- Maximal/minimal for identity:
  - For similarity measures: $\forall i : d(i, i) = max_k d(i, k)$
  - For dissimilarity measures: $\forall i : d(i, i) = 0$
- A distance metric is a dissimilarity metric that satisfies the triangle inequality

$$\forall i, j, k : d(i, j) + d(i, k) \geq d(j, k)$$

- Distance metrics are non-negative: $\forall i, k : d(i, k) \geq 0$

Similarity measures, binary

X is the set of all terms occurring in document D_X, Y is the set of all terms occurring in document D_Y.

- Raw Overlap: $raw\_overlap(X, Y) = |X \cap Y|$
- Dice's coefficient: (normalisation by average size of the two original vectors)

$$dice(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

- Jaccard's coefficient: (normalisation by size of combined vector – penalises small number of shared feature values)

$$jacc(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

- Overlap coefficient:

$$overlap\_coeff(X, Y) = \frac{|X \cap Y|}{min(|X|, |Y|)}$$

- Cosine: (normalisation by vector lengths)

$$cosine(X, Y) = \frac{|X \cap Y|}{\sqrt{|X|} \cdot \sqrt{|Y|}}$$

Similarity measures, weighted

Weighted versions of Dice's and Jaccard's coefficient exist, but are used rarely for IR:

- Vectors are extremely sparse
- Vectors are of very differing length

Cosine (or normalised inner product) is the measure of choice for IR

Document i is represented as a vectors of terms or lemmas ($\vec{w_i}$); t is the total number of index terms in system, $w_{i,j}$ is the weight associated with j th term of vector $\vec{w_i}$.

Vector length normalisation by the two vectors $|\vec{w_i}|$ and $|\vec{w_k}|$:

$$cos(\vec{w_i}, \vec{w_k}) = \frac{\vec{w_i}\vec{w_k}}{|\vec{w_i}| \cdot |\vec{w_k}|} = \frac{\Sigma_{j=1}^{d} w_{i,j} \cdot w_{k,j}}{\sqrt{\Sigma_{j=1}^{d} w_{i,j}^2} \cdot \sqrt{\Sigma_{j=1}^{d} w_{k,j}^2}}$$

Distance measures

- Euclidean distance: (how far apart in vector space)

$$euc(\vec{w_i}, \vec{w_k}) = \sqrt{\sum_{j=1}^{d} (w_{i,j} - w_{k,j})^2}$$

- Manhattan distance: (how far apart, measured in 'city blocks')

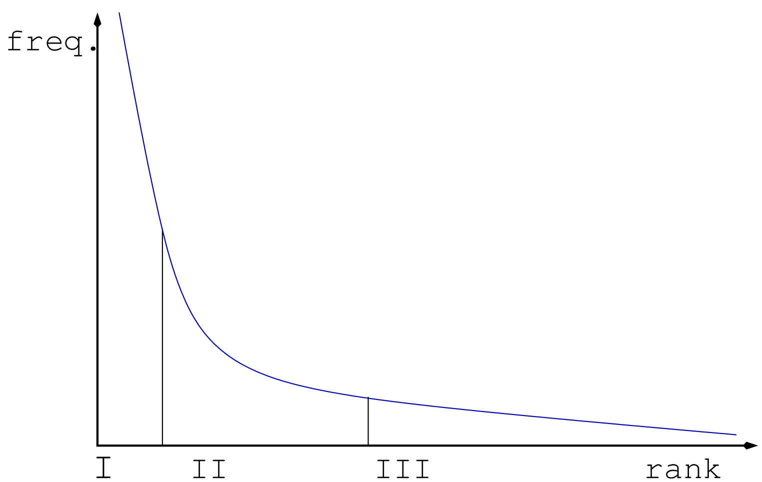
$$manh(\vec{w_i}, \vec{w_k}) = \sum_{j=1}^{d} |w_{i,j} - w_{k,j}|$$

Zipf's law: the rank of a word is reciprocally proportional to its frequency:

$$freq(word_i) = \frac{1}{i^\theta} freq(word_1)$$

(with $1.5 < \theta < 2$ for most languages)

($word_i$ being the ith most frequent word of the language)



- Zone I: High frequency words tend to be functional words ("the", "of")
- Zone III: Low frequency words tend to be typos, or unimportant words (too specific) ("Uni7ed", "super-noninteresting", "87-year-old", "0.07685")
- Zone II: Mid-frequency words are the best indicators of what the document is about

Term Weighting: TF*IDF

Not all terms describe a document equally well:

- Terms which are frequent in a document are better → $tf_{w,d} = freq_{w,d}$ should be high
- Terms that are overall rare in the document collection are better → $idf_{w,D} = log\frac{|D|}{n_{w,D}}$ should be high →
- TF*IDF formula: $tf*idf_{w,d,D} = tf_{w,d} \cdot idf_{w,D}$ should be high
- Improvement: Normalise $tf_{w,d}$ by term frequency of most frequent term in document: $tf_{norm,w,d} = \frac{freq_{w,d}}{max_{l \in d} freq_{l,d}}$
  - Normalised TF*IDF: $tf * idf_{norm,w,d,D} = tf_{norm,w,d} \cdot idf_{w,D}$

$tf_{w,d}$:	Term frequency of word w in document d
$n_{w,D}$:	Number of documents in document collection D which contain word w
$idf_{w,D}$:	Inverse document frequency of word w in document collection D
$tf * idf_{w,d,D}$:	TF*IDF weight of word w in document d in document collection D
$tf * idf_{norm,w,d,D}$:	Length-normalised TF*IDF weight of word w in document d in document collection D
$tf_{norm,w,d}$:	Normalised term frequency of word w in document d
$max_{l \in d} freq_{l,d}$:	Maximum term frequency of any word in document d

Example: TF*IDF

Document set: 30,000

Term	tf	$n_{w,D}$	TF*IDF
the	312	28,799	5.55
in	179	26,452	9.78
general	136	179	302.50
fact	131	231	276.87
explosives	63	98	156.61
nations	45	142	104.62
1	44	2,435	47.99
haven	37	227	78.48
2-year-old	1	4	3.88

IDF(“the”) = log $(\frac{30{,}000}{28{,}799})$ = 0.0178

TF*IDF(“the”) = 312 $\cdot$ 0.0178 = 5.55

Example: VSM (TF*IDF; cosine)

	Q	D_{7655}	D_{454}
hunter	19.2	56.4	112.2
gatherer	34.5	122.4	0
Scandinavia	13.9	0	30.9
30,000	0	457.2	0
years	0	12.4	0
BC	0	200.2	0
prehistoric	0	45.3	0
deer	0	0	23.6
rifle	0	0	452.2
Mesolithic	0	344.2	0

$$\cos(Q, D_{7655}) = \frac{19.2 \cdot 56.4 + 34.5 \cdot 122.4 + 13.9 \cdot 0}{\sqrt{19.2^2 + 34.5^2 + 13.9^2} \cdot \sqrt{56.4^2 + 122.4^2 + 457.2^2 + 12.4^2 + 200.2^2 + 45.3^2 + 344.2^2}} = .2037698341$$

$$\cos(Q, D_{454}) = \frac{19.2 \cdot 112.2 + 34.5 \cdot 0 + 13.9 \cdot 30.9}{\sqrt{19.2^2 + 34.5^2 + 13.9^2} \cdot \sqrt{112.2^2 + 30.9^2 + 23.6^2 + 452.2^2}} = .1322160530$$

$\longrightarrow$ choose document D_{7655}

Self test VSM/ TF*IDF

- Build a document-term matrix for three (very!) short documents of your choice
- Weight by presence/absence (binary) and by TF*IDF (with estimated IDFs)
- Write a suitable query
- Calculate document–query similarity, using
  - cosine
  - inner product (i.e. cosine without normalisation)
- What effect does normalisation have?

Term Manipulation

- So far: each term is indexed and weighted only in string-equal form
- This misses many semantic similarities between morphologically related words (“whale” $\rightarrow$ “whaling”, “whales”)
- Automatic models of term identity
  - The same string between blanks or punctuation
  - The same prefix (eg. up to 6 characters)
  - The same stem (e.g. Porter stemmer)
  - The same linguistic lemma (sensitive to Parts-of-speech)
- Effect of term manipulation on retrieval result
  - changes the counts, reduces total number of terms
  - increases recall
  - might decrease precision, introduction of noise

Stemming: the Porter stemmer

M. Porter, “An algorithm for suffix stripping”, Program 14(3):130-137, 1980

CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS

- Removal of suffixes without a stem dictionary, only with a suffix dictionary
- Terms with a common stem have similar meanings:
- Deals with inflectional and derivational morphology
- Conflates relate — relativity — relationship
- Treats Sand — sander and wand — wander the same (does not conflate either, though sand/sander arguably could be conflated)
- Root changes (deceive/deception, resume/resumption) aren’t dealt with, but these are rare

$$[C]\ (VC)\{m\}[V]$$

C	one or more adjacent consonants
V	one or more adjacent vowels
[ ]	optionality
( )	group operator
{x}	repetition x times
m	the "measure" of a word

shoe	$[sh]_C[oe]_V$	m=0
Mississippi	$[M]_C([i]_V[ss]_C)([i]_V[ss]_C)([i]_V[pp]_C)[i]_V$	m=3
ears	$([ea]_V[rs]_C)$	m=1

Notation: m is calculated on the word excluding the suffix of the rule under consideration (eg. In m=1 for 'element' in rule "(m $>$ 1) EMENT", so this rule would not trigger.)

Porter stemmer: rules and conditions

Rules in one block are run through in top-to-bottom order; when a condition is met, execute rule and jump to next block

Rules express criteria under which suffix may be removed from a word to leave a valid stem: (condition) S1 $\rightarrow$ S2

Possible conditions:

- constraining the measure:

  (m $>$ 1) EMENT $\rightarrow \epsilon$ (ϵ is the empty string)

  REPLACEMENT $\rightarrow$ REPLAC

- constraining the shape of the word piece:
  - *S – the stem ends with S
  - *v* – the stem contains a vowel
  - *d – the stem ends with a double consonant (e.g. -TT, -SS).
  - *o – the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP)

- expressions with AND, OR and NOT:
  - (m$>$1 AND (*S OR *T)) – a stem with m> 1 ending in S or T

Porter stemmer: selected rules

SSES → SS IES → I SS → SS S →

caresses → caress
cares → care

(m>0) EED → EE

feed → feed
agreed → agree
BUT: freed, succeed

(*v*) ED →

plastered → plaster
bled → bled

Porter stemmer: the algorithm

Step 1: plurals and past participles

Step 1a

SSES	$\rightarrow$ SS	caresses	$\rightarrow$ caress
IES	$\rightarrow$ I	ponies	$\rightarrow$ poni
		ties	$\rightarrow$ ti
SS	$\rightarrow$ SS	caress	$\rightarrow$ caress
S	$\rightarrow \epsilon$	cats	$\rightarrow$ cat

Step 1b

(m>0)	EED	$\rightarrow$ EE	feed	$\rightarrow$ feed
			agreed	$\rightarrow$ agree
(*v*)	ED	$\rightarrow \epsilon$	plastered	$\rightarrow$ plaster
			bled	$\rightarrow$ bled
(*v*)	ING	$\rightarrow \epsilon$	motoring	$\rightarrow$ motor
			sing	$\rightarrow$ sing

If rule 2 or 3 in Step 1b applied, then clean up:

AT	$\rightarrow$ ATE	conflat(ed/ing)	$\rightarrow$ conflate
BL	$\rightarrow$ BLE	troubl(ed/ing)	$\rightarrow$ trouble
IZ	$\rightarrow$ IZE	siz(ed/ing)	$\rightarrow$ size
(*d and not (*L or *S or *Z))	$\rightarrow$ single letter	hopp(ed/ing)	$\rightarrow$ hop
		hiss(ed/ing)	$\rightarrow$ hiss
(m=1 and *o)	$\rightarrow$ E	fil(ed/ing)	$\rightarrow$file
		fail(ed/ing)	$\rightarrow$ fail

Step 1c

(*v*)	Y	$\rightarrow$ I	happy	$\rightarrow$ happi
			sky	$\rightarrow$ sky

Step 2: derivational morphology

(m>0)	ATIONAL	→ ATE	relational	→ relate
(m>0)	TIONAL	→ TION	conditional	→ condition
			rational	→ rational
(m>0)	ENCI	→ ENCE	valenci	→ valence
(m>0)	ANCI	→ ANCE	hesitanci	→ hesitance
(m>0)	IZER	→ IZE	digitizer	→ digitize
(m>0)	ABLI	→ ABLE	conformabli	→ conformable
(m>0)	ALLI	→ AL	radicalli	→ radical
(m>0)	ENTLI	→ ENT	differentli	→ different
(m>0)	ELI	→ E	vileli	→ vile
(m>0)	OUSLI	→ OUS	analogousli	→ analogous
(m>0)	IZATION	→ IZE	vietnamization	→ vietnamize
(m>0)	ATION	→ ATE	predication	→ predicate
(m>0)	ATOR	→ ATE	operator	→ operate
(m>0)	ALISM	→ AL	feudalism	→ feudal
(m>0)	IVENESS	→ IVE	decisiveness	→ decisive
(m>0)	FULNESS	→ FUL	hopefulness	→ hopeful
(m>0)	OUSNESS	→ OUS	callousness	→ callous
(m>0)	ALITI	→ AL	formaliti	→ formal
(m>0)	IVITI	→ IVE	sensitiviti	→ sensitive
(m>0)	BILITI	→ BLE	sensibiliti	→ sensible

Step 3: more derivational morphology

(m>0)	ICATE →	IC	triplicate	→ triplic
(m>0)	ATIVE →	ϵ	formative	→ form
(m>0)	ALIZE →	AL	formalize	→ formal
(m>0)	ICITI →	IC	electriciti	→ electric
(m>0)	ICAL →	IC	electrical	→ electric
(m>0)	FUL →	ϵ	hopeful	→ hope
(m>0)	NESS →	ϵ	goodness	→ good

Step 4: even more derivational morphology

(m>1)	AL →	ϵ	revival	→ reviv
(m>1)	ANCE →	ϵ	allowance	→ allow
(m>1)	ENCE →	ϵ	inference	→ infer
(m>1)	ER →	ϵ	airliner	→ airlin
(m>1)	IC →	ϵ	gyroscopic	→ gyroscop
(m>1)	ABLE →	ϵ	adjustable	→ adjust
(m>1)	IBLE →	ϵ	defensible	→ defens
(m>1)	ANT →	ϵ	irritant	→ irrit
(m>1)	EMENT →	ϵ	replacement	→ replac
(m>1)	MENT →	ϵ	adjustment	→ adjust
(m>1)	ENT →	ϵ	dependent	→ depend
(m>1 and (*S or *T))	ION →	ϵ	adoption	→ adopt
(m>1)	OU →	ϵ	homologou	→ homolog
(m>1)	ISM →	ϵ	communism	→ commun
(m>1)	ATE →	ϵ	activate	→ activ
(m>1)	ITI →	ϵ	angulariti	→ angular
(m>1)	OUS →	ϵ	homologous	→ homolog
(m>1)	IVE →	ϵ	effective	→ effect
(m>1)	IZE →	ϵ	bowdlerize	→ bowdler

Step 5: cleaning up

Step 5a

(m>1)	E → ϵ	probate	→ probat
		rate	→ rate
(m=1 and not *o)	E → ϵ	cease	→ ceas

Step 5b

(m > 1 and *d and *L)	→ single letter	controll	→ control
		roll	→ roll

Self test Porter Stemmer

1. Show which stems *rationalisations, rational, rationalizing* result in, and which rules they use.
2. Explain why *sander* and *sand* do not get conflated.
3. What would you have to change if you wanted to conflate them?
4. Find five different examples of incorrect stemmings.
5. Can you find a word that gets reduced in every single step (of the 5)?
6. Exemplify the effect that stemming (eg. with Porter) has on the Vector Space Model, using your example from before.

Summary and literature

- Indexing languages
- Retrieval models
- Term weighting
- Term stemming

Textbook (Baeza-Yates and Ribeiro-Neto):

- 2.5.2 Boolean model
- 6.3.3 Zipf's law
- 2.5.3 Vector space model, TF*IDF
- 7.2 Term manipulation, stemming