EXAMPLE

Let the set of original documents (to be searched) be

$$D = \{D_1, D_2, D_3\},$$

where

D_1 = Bayes' Principle: The principle that, in estimating a parameter, one should initially assume that each possible value has equal probability (a uniform prior distribution).

D_2 = Bayesian Decision Theory: A mathematical theory of decision making which presumes utility and probability functions, and according to which the act to be chosen is the Bayes act, i.e. the one with highest Subjective Expected Utility. If one had unlimited time and calculating power with which to make every decision, this procedure would be the best way to make any decision.

D_3 = Bayesian Epistemology: A philosophical theory which holds that the epistemic status of a proposition (i.e. how well proven or well established it is) is best measured by a probability and that the proper way to revise this probability is given by Bayesian conditionalisation or similar procedures. A Bayesian epistemologist would use probability to define, and explore the relationship between, concepts such as epistemic status, support or explanatory power.

TASK: process and prepare *D* for retrieval.

SOLUTION: apply *IR* technology.

Steps

1. **Identify lexical units.** Write a computer program to recognise words (word = any sequence of characters preceded and followed by ‘space’, ‘dot’, ‘comma’). For example: probability, Bayesian, epistemology.

2. **Stoplisting.** Create a list of ranked words (Power Law). Exclude frequent and rare words (use threshold values).
3. **Stemming.** Apply stemming to remaining words. The thus obtained words are called ***index terms*** (because they are used to identify documents), and form a set ***T***.

Original text:

```
Bayes' Principle: The principle that, in estimating a
parameter, one should initially assume that each
possible value has equal probability (a uniform prior
distribution).
```

```
function processline($line,$outputfile)
{
  $elements=split("[^A-Za-z0-9]",$line);
  foreach ($elements as $e)
  if (trim($e)!='')
     fwrite($outputfile,strtolower(trim($e))."\r\n");
}
```

Output:

bayes	assume
principle	that
the	each
principle	possible
that	value
in	has
estimating	equal
a	probability
parameter	a
one	uniform
should	prior
initially	distribution

Stoplist:

a	afterwards
aboard	again
about	against
above	ago
accordingly	all
across	allows
actually	almost
add	alone
added	along
after	alongside
	...

The Porter Stemming Algorithm

In Linguistics, a **morpheme** is the smallest meaningful unit in a given language.

A **suffix** is a morpheme that is attached to at the back of a base morpheme to form a word.

Ex.: connect + ion=connection

The Porter stemming algorithm (or 'Porter stemmer') is a process for removing the suffixes from words in English.

Ex.: connection → connect

Complex suffixes are removed step by step. Thus: GENERALIZATIONS is stripped to GENERALIZATION (Step 1), then to GENERALIZE (Step 2), then to
GENERAL (Step 3), and then to
GENER (Step 4).

The *rules* for removing a suffix are given in the form:

(condition) S1 -> S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2.

Ex #1.:

(*S or *T) ION -> .

Here S1 is `ION' and S2 is null. This would map ADOPTION to ADOPT because the word ends with letter S or T.

Rules:

```
SSES -> SS          caresses  ->  caress
IES  -> I           ponies    ->  poni
                    ties      ->  ti
SS   -> SS          caress    ->  caress
S    ->             cats      ->  cat

ATIONAL->ATE     relational  -> relate
TIONAL ->TION    conditional -> condition
                 rational    -> rational
IZER   ->IZE     digitizer   -> digitize
ATION  ->ATE     predication -> predicate
```

Stemming Hungarian

The stemmer is based on a spell checker which has rules like:

Rule:	**Example:**
Z -> Z	**ráz -> rázz**
Z -> ZÁL	**ráz -> rázzál**
E D Z -> ENEK	**edz -> edzenek, pedz -> pedzenek**
[ÓUÚ] -> VAL	**manó -> manóval, hamu -> hamuval, bú -> búval**
[^AE] -> NAK	**okos -> okosnak**
. -> KÉNT	**kutya -> kutyaként, okos -> okosként**
A -> -A,ÁNAK	**macska -> macskának**
[^SD] ZIK -> -IK,Z	**barátkozik -> barátkozz**

Text before and	after stemmig:
bayes	bay
principle	principl
the	the
principle	principl
that	that
in	in
estimating	estim
a	a
parameter	paramet
one	on
should	should
initially	initi
assume	assum
that	that
each	each
possible	possibl
value	valu
has	ha
equal	equal
probability	probabl
a	a
uniform	uniform
prior	prior
distribution	distribut

Stoplist before and	**after stemming:**
a	a
aboard	aboard
about	about
above	abov
accordingly	accordingli
across	across
actually	actual
add	ad
added	add
after	after
afterwards	afterward
again	again
against	against
ago	ago
all	all
allows	allow
almost	almost
alone	alon
along	along
alongside	alongside
...	...

Original text:

```
Bayes' Principle: The principle that, in estimating a
parameter, one should initially assume that each
possible value has equal probability (a uniform prior
distribution).
```

After stemming and removing stopwords:

```
bay
principl
principl
estim
paramet
initi
assum
equal
uniform
prior
distribut
```

What if the source is an HTML document?

```
2004.  &#233;vi  felv&#233;teli  jelentkez&#233;si
adatok, Gazdas&#225;gtudom&#225;nyi kar
```

One method to resolve this:

```
$string = str_replace ( array('&#225;','&aacute;'),
'á', $string );


$string = str_replace ( array('&#193;','&Aacute;'),
'Á', $string );


$string = str_replace (
array('&ucirc;','&#251;','&#369;','&#305;'), 'ű',
$string );
```

Let the set T of index terms be (not stemmed here):

$T = \{t_1, t_2, t_3\} = \{t_1 = \text{Bayes}, t_2 = \text{probability}, t_3 = \text{epistemology}\}.$

4. Build term-document matrix.

Conceive the documents as sets of terms together with their frequencies):

$D_1 = \{$ (Bayes, 1); (probability, 1); (epistemology, 0) $\}$

$D_2 = \{$ (Bayes, 2); (probability, 1); (epistemology, 0) $\}$

$D_3 = \{$ (Bayes, 3); (probability, 3); (epistemology, 3) $\}$

Term-document matrix: *TD*

$TD_{3\times3} = (w_{ij})$, where w_{ij} denotes the weight of term t_i in document D_j

Frequency: $TD = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 3 \\ 0 & 0 & 3 \end{pmatrix}$

Binary: $TD = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Uncompressed Matrix

A=

11	**0**	**0**	**14**	**0**	**0**
0	**22**	**0**	**0**	**0**	**0**
0	**0**	**33**	**0**	**0**	**36**
0	**0**	**0**	**44**	**0**	**0**
0	**0**	**0**	**0**	**0**	**56**
0	**0**	**0**	**0**	**0**	**66**

Compressed format

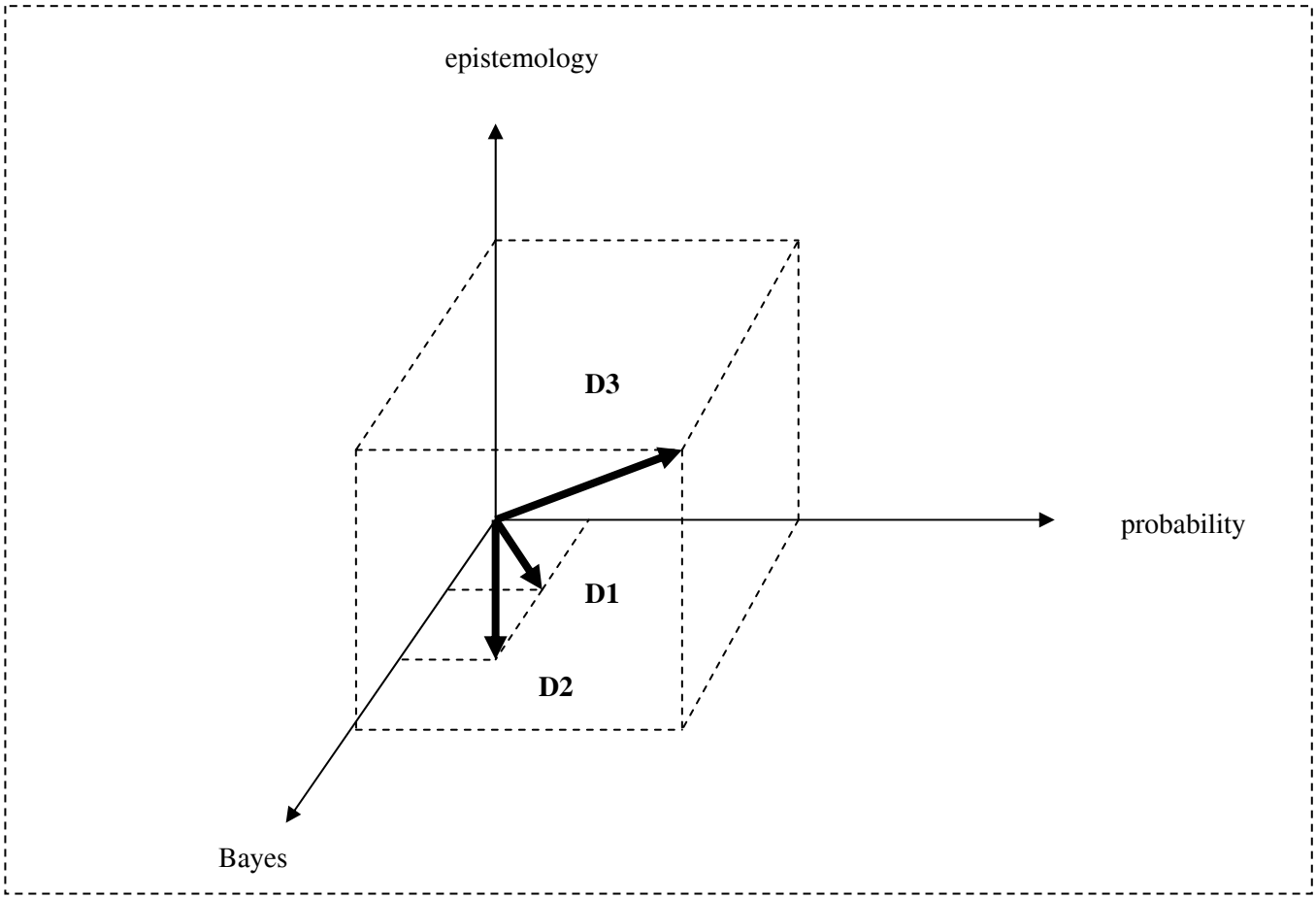
Row	Coloumn	Value
1	**1**	**11**
1	**4**	**14**
2	**2**	**22**
3	**3**	**33**
3	**6**	**36**
4	**4**	**44**
5	**6**	**56**
6	**6**	**66**

Length Normalisation

Take the frequency $TD = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 3 \\ 0 & 0 & 3 \end{pmatrix}$

Represent documents in the space of terms:

- use a system of coordinate axes to represent documents
- each term t_i corresponds to a coordinate axis
- every document D_j will be a vector $\mathbf{D}j$ having coordinates $\mathbf{D}_j = (f_{1j}, f_{2j}, f_{3j})$



We can see that document vectors have different lengths.

Thus, we may define the notion of ***length of document***.
Document length is a quantity which is proportional to the number of terms the document contains.
Hence, longer documents will 'dominate' shorter ones.
Unfair!
Why should a wordy writer (who says everything five times in five different ways) be 'preferred' to a writer who has worked hard to find the most appropriate words to express something?

One way to overcome this problem is **normalisation**.

There are several normalisation methods.

Maximum normalisation:

weights on 0-1 scale
relative to itself

$$TD = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0.5 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Length normalisation:

weights on 0-1 scale
every document vector will have unit length

For example:

vector $\mathbf{D}_1 = (1\ 1\ 0)$

length of vector $\mathbf{D}_1 = (1^2 + 1^2 + 0^2)^{0.5} = \sqrt{2}$

weights of new vector $\mathbf{D}_1 = (1/\sqrt{2} \quad 1/\sqrt{2} \quad 0/\sqrt{2})$

length of new vector $\mathbf{D}_1 = ((1/\sqrt{2})^2 + (1/\sqrt{2})^2 + 0^2)^{0.5} =$

$$(1/2 + 1/2 + 0)^{0.5} = 1$$

New term-document matrix (with length-normalised weights):

$$TD = \begin{pmatrix} 1/\sqrt{2} & 2/\sqrt{5} & 3/\sqrt{27} \\ 1/\sqrt{2} & 1/\sqrt{5} & 3/\sqrt{27} \\ 0 & 0 & 3/\sqrt{27} \end{pmatrix}$$