Advanced Text Methods

Matt Perdeaux

matt@associativetrails.com

Associative Trails Ltd.

www.associativetrails.com

Plan

Introduction to the theory behind Latent Semantic Indexing - one method of text analysis & retrieval.

We will look at:

- A list of different methods of text search and retrieval
- The term space - the building block of LSI
- The use of vectors to mathematically model “documents”
- Steps taken to build document vectors and look at how similarity is calculated
- Tuning the term space
- Further reading

Different approaches

- SQL character matching - “coldfus%” and regular expressions
- Advanced pattern matching - produces “rank” via proximity (for multiple terms), frequency, importance.
- Phonetic algorithms (e.g. Soundex) - copes with small variations between words - misspellings and typos. E.g. “Search” = S620 = “Serach”
- Bayesian filtering - statistical techniques based on a frequency interpretation of probability. Groups similar “documents” together into categories - used by Autonomy and proving very effective in spam filtering.
- Latent Semantic Indexing (LSI) - a natural language processing technique that builds a mathematical model of the documents in a collection and analyses them using vector and matrix algebra

Latent Semantic Indexing

- Created in 1990 - “Indexing by Latent Semantic Analysis”, Deerswester et al, *Journal of the Society for Information Science*
- Attempts to solve two fundamental problems: **synonymy** (using different words to describe the same idea) and **polysemy** (same word having multiple meanings)
- Does not use boolean logical operators
- Ideal for long queries or QBE (query by example) functionality - looks at collections of documents
- Based on an n-dimensional vector space

Example “Documents”

Document A

Kevin Pietersen smashed his first ever Test century to earn the draw that gave England a 2-1 series win and the Ashes for the first time since 1987. Pietersen, who was dropped twice, hit 158 and Ashley Giles 59 before the English team were bowled out for 335 at The Oval, to snuff out Australian hopes of victory.

Document B

Kevin Pietersen stole the show to guide England home by three wickets as Australia were dealt a second NatWest Series defeat in two days. Australia, humiliated by Bangladesh on Saturday, set England 253 to win in Bristol and looked on top when reducing them to 160-6 in the 38th over.

Analyse Two Terms

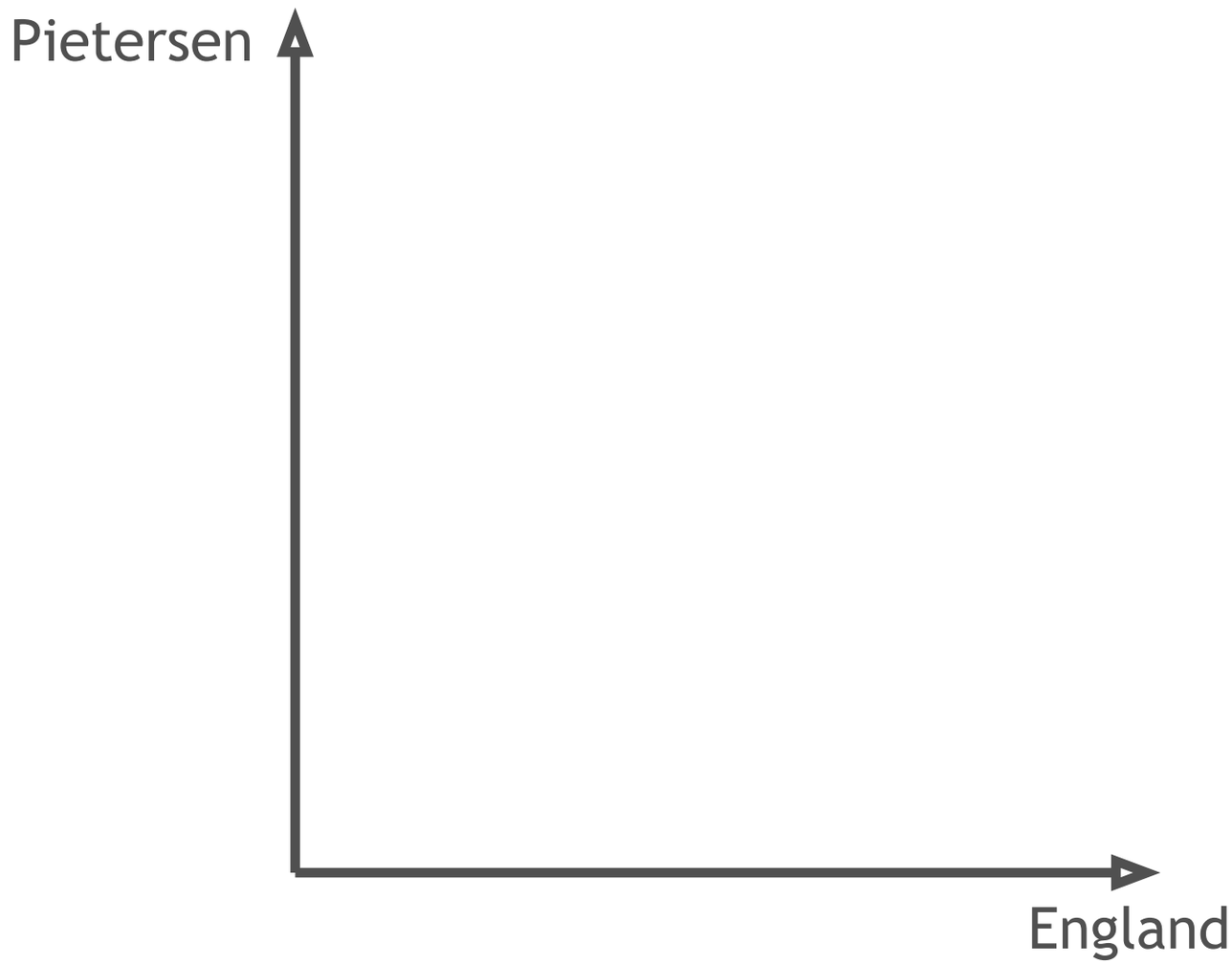
Document A

Kevin **Pietersen** smashed his first ever Test century to earn the draw that gave **England** a 2-1 series win and the Ashes for the first time since 1987. **Pietersen**, who was dropped twice, hit 158 and Ashley Giles 59 before the English team were bowled out for 335 at The Oval, to snuff out Australian hopes of victory.

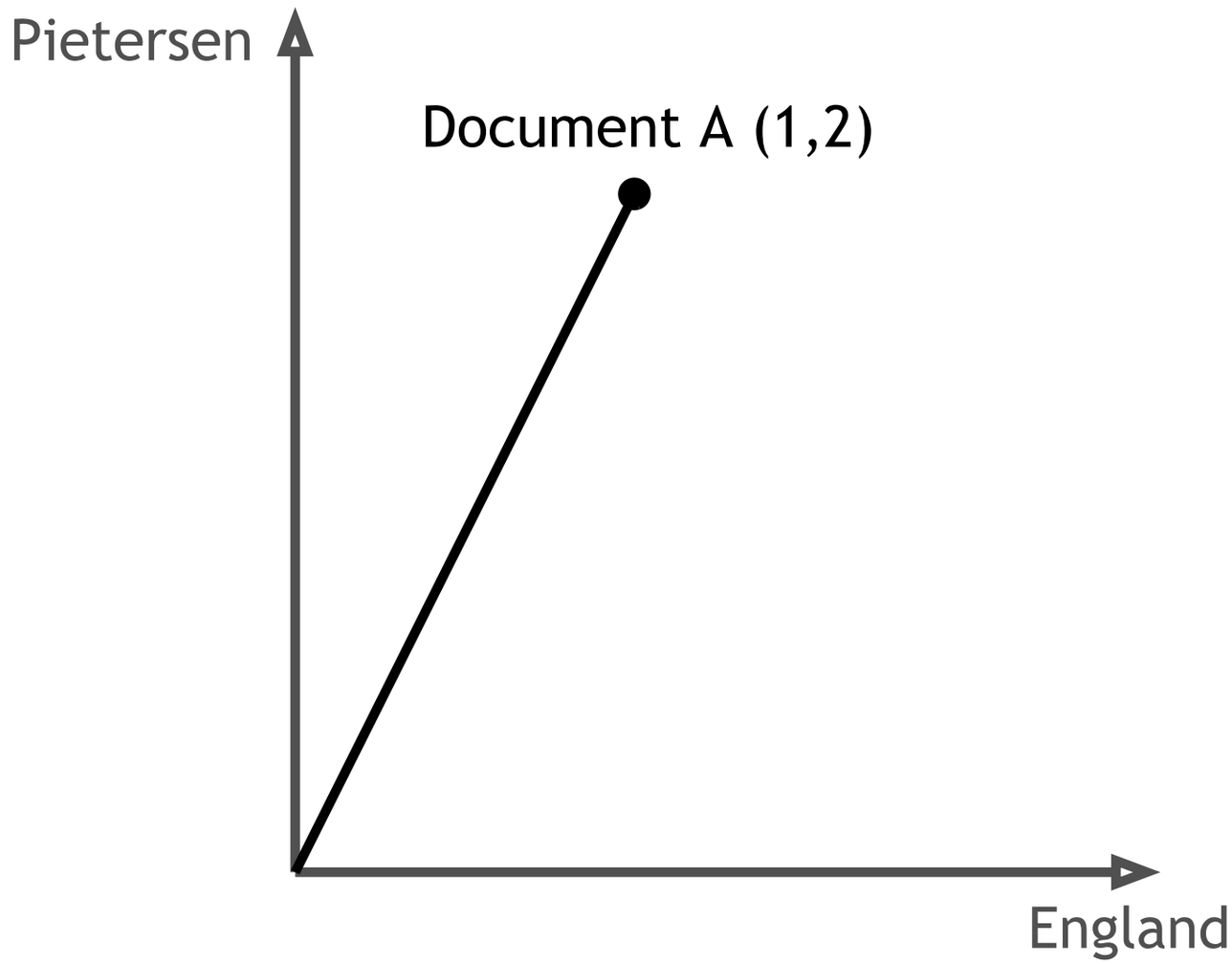
Document B

Kevin **Pietersen** stole the show to guide **England** home by three wickets as Australia were dealt a second NatWest Series defeat in two days. Australia, humiliated by Bangladesh on Saturday, set **England** 253 to win in Bristol and looked on top when reducing them to 160-6 in the 38th over.

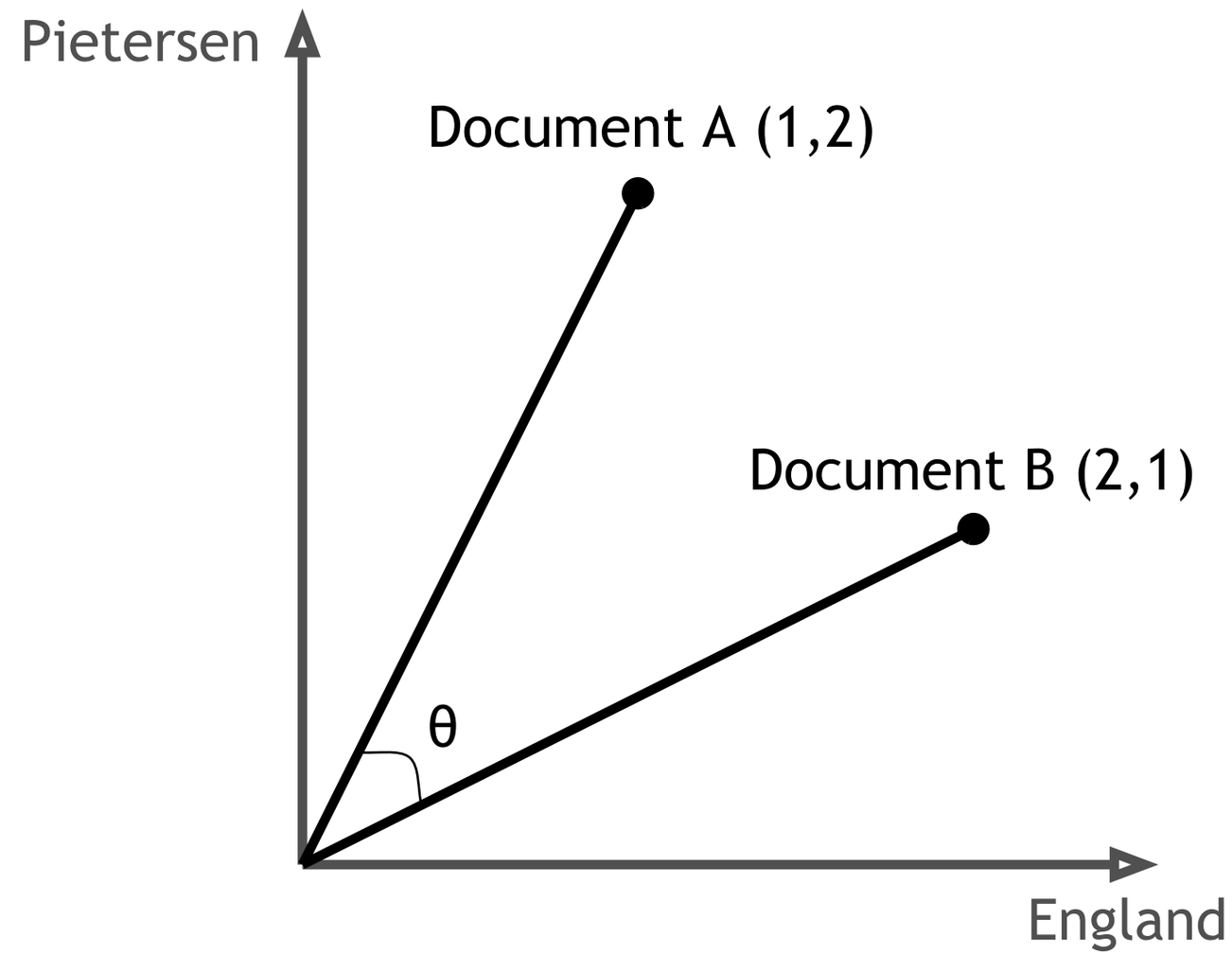
Build Term Space - Axes



Build Term Space - Vector 1



Build Term Space - Vector 2



Calculating the similarity

To find the angle between two vectors, use the scalar (or dot) product:

$$\cos\theta = \frac{A \,.\, B}{|A| \,.\, |B|}$$

where **A = 1i + 2j** and **B = 2i + 1j**

So,

$$\cos\theta = \frac{(1i.2i)+(2j.1j)}{\sqrt{(1^2+2^2)}.\sqrt{(2^2+1^2)}} = \frac{(1.2)+(2.1)}{\sqrt{5}.\sqrt{5}} = \frac{4}{5} = \frac{0.8}{}$$

Therefore, based on our two chosen terms, the documents show an **80% similarity** (that's **very** close).

Refining the item vectors

- Strip punctuation and markup - regex
- Remove “stop” words - common articles, pronouns, etc. as they do not convey meaning. Dependent on context - a CF blog might want to add the words “coldfusion” & “macromedia” to the stop word list.
- Stemming - “merges” different forms of the same word - singular and plural, verb tenses, etc.

Apply weightings to smooth out various effects in the global model:

- Term frequency
- Collection frequency weight
- Document length normalisation
- Combined Weight

Strip Punctuation

Document A

kevin pietersen smashed his first ever test century to earn the draw that gave england a series win and the ashes for the first time since pietersen who was dropped twice hit and ashley giles before the english team were bowled out for at the oval to snuff out australian hopes of victory

Document B

kevin pietersen stole the show to guide england home by three wickets as australia were dealt a second natwest series defeat in two days australia humiliated by bangladesh on saturday set england to win in bristol and looked on top when reducing them to in the over

Remove “Stop” Words

Document A

kevin pietersen smashed first ever test century earn draw gave england series win ashes first time pietersen dropped twice hit ashley giles before english team bowled oval snuff australian hopes victory

Document B

kevin pietersen stole show guide england home three wickets australia dealt second natwest series defeat two days australia humiliated bangladesh saturday england win bristol looked top reducing over

Stemming

- A stemmer is an algorithm that reduces similar words to a common root or “stem”
- Stemming algorithms are quite straightforward in english
- The most widely used algorithm is by Martin Porter - initially published in 1980 with more recent revisions. http://www.tartarus.org/martin/PorterStemmer/.
- e.g. “fishing”, “fished”, “fish” & “fisher” => “fish”

Stemmed Output

Document A

kevin pietersen smashe first ever test centuri earn draw gave england seri win ash first time pietersen drop twice hit ashlei gile befor english team bowle oval snuff australia hope victori

Document B

kevin pietersen stole show guide england home three wicket australia dealt second natwest seri defeat two dai australia humili bangladesh saturdai england win bristol looke top reduc over

Term Frequencies

Term	Frequency(1)	Frequency(2)
pietersen	2	1
australia	1	2
england	1	2
win	1	1
seri	1	1
kevin	1	1
first	2	0
dai	0	1
centuri	1	0
bristol	0	1
…	…	…
ash	1	0

Calculate Dot Product

So, calculating the dot product:

$$\cos\theta = \frac{A \cdot B}{|A| \cdot |B|}$$

$$\cos\theta = \frac{(2x1)+(1x2)+(1x2)+(1x1)+(1x1)+...+(1x0)}{\sqrt{(2^2+1^2+1^2+1^2+1^2..1^2)} x \sqrt{(1^2+2^2+2^2+1^2+1^2..0^2)}}$$

$$\cos\theta = \frac{(\ 9\ )}{\sqrt{35}.\sqrt{32}} = 0.273 = 27.3\% \text{ similarity}$$

Notes on the calculation

- Similarity - “only” 27% for two documents on the same subject and of similar length
- Numbers along the top - of the 49 pairs of values, only 6 of them actually required calculation. Any term with a zero frequency can be ignored in the calculations.
- The numbers on the bottom are the lengths of the vectors (Pythagoras theorem). These are individual to the document and can be cached in the persistence layer when the item is first added.

Tuning the model

Collection frequency weight (CFW) - factors a particular term based on how many documents it occurs in. The more documents the term occurs in, the lower its influence should be. Robertson & Sparck Jones suggests:

= logN - logn

where, n = no. of documents in which term occurs
and N = no. of documents

Further Tuning

Document Length Normalisation (NDL) - tries to even out the effect of having some document that are much longer (and have many more terms) than others. Robertson & Sparck Jones suggests:

= DL/average DL.

where, DL = any measure of document length (e.g. no. of characters, number of stemmed terms, etc.)

Further Tuning (cont.)

Combining the various weighting terms we already have: TF (term frequency), CFW (collection frequency weight) & NDL (normalisd document length). Robertson & Sparck Jones suggests:

$$CW = \frac{CFW \times TF \times (k+1)}{k \times ((1-b)+(b \times NDL)+TF)}$$

where, k = influence of TF (try 2)
and b = influence of document length 0<b<1
when b=0, long documents are multitopic
b=1, long documents are verbose
try b=0.75

Next Steps

- Building the term space is only the first step in “proper” LSI.
- The next step is really scary - Singular Value Decomposition (SVD).
- An algorithm that approximates a matrix - a mathematical way of reducing the number of axes (terms) and still retaining the overall approximate shape of the matrix (or vector space).
- This has two major advantages: it reduces the number of calculations and it squashes similar terms together. This is the really powerful part of LSI.
- Incredibly complicated and way beyond my A-level maths...!

References

- **Indexing by Latent Semantic Analysis**. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman (1990) *Journal of the Society for Information Science*, 41(6), 391-407. http://lsi.research.telcordia.com/lsi/papers/JASIS90.pdf
- **An Algorithm for Suffix Stripping**. Porter, M.F. (1980). *Program*, 14(3): 130-137. http://www.tartarus.org/martin/PorterStemmer/def.txt. http://www.tartarus.org/martin/PorterStemmer/.
- **Simple, proven approaches to text retrieva**l. Robertson, S.E. & Sparck Jones, K. (1997). Technical Report TR356, University of Cambridge, Computer Laboratory. http://www.cl.cam.ac.uk/Research/Reports/TR356-ksj-approaches-to-text-retrieval.html
- **Computational Methods for Intelligent Information Access.** W. Berry, S. T. Dumais & T. A. Letsche (1995). *Proceedings of Supercomputing '95, San Diego, CA*. http://www.cs.utk.edu/~berry/sc95/sc95.html

And not forgetting...

- **Building a Keyword Vector Space Engine in Coldfusion**. M. Perdeaux (2004) *Coldfusion Developers Journal*, 6(8), 42-50. http://coldfusion.sys-con.com/read/45980.htm