# Recuperación de Información

## *2.4 Recuperación de Información Textual Avanzada*
## *2.5 Aspectos relacionados con la recuperación de información textual*

David E. Losada

{dlosada}@dec.usc.es

Grupo de Sistemas Inteligentes

Departamento de Electrónica y Computación

Universidad de Santiago de Compostela

# Outline

- Query modification and relevance feedback
- Clusterización
- Cross Language Information Retrieval

# Query modification and RF

- Relevance feedback (RF)
  - Modificar automáticamente la consulta de usuario
  - Por qué es útil?
  - Cómo hacerlo en distintos modelos de RI?

# Relevance feedback

- Consulas difíciles de crear. Los usuarios usalmente no saben exactamente lo que están buscando.
- Las consultas suelen ser expresiones muy burdas de una necesidad de información.
- Expresar de modo exacto lo que queremos puede ser difícil
  - Frecuentemente es más fácil reconocer qué información es relevante

# Relevance feedback

- Mostrarle al sistema qué es lo que quieres
- El sistema dispone pues de ejemplos de documentos relevantes
- El sistema modifica automáticamente tu consulta

# Relevance feedback

- Los docs relevantes a una consulta se suelen parecer entre sí
- Si disponemos de algunos docs relevantes será más fácil buscar otros que también lo sean.
- Detectar palabras o términos útiles. Nuevos términos para la consulta (expansión de la consulta)
- Detectar cuán útiles son los términos. Cambiar los pesos de las palabras en la consulta original (*term reweighting*), p.e. en el modelo vectorial.
- Usar la nueva consulta para un nuevo proceso de recuperación.

# Expansión de la consulta

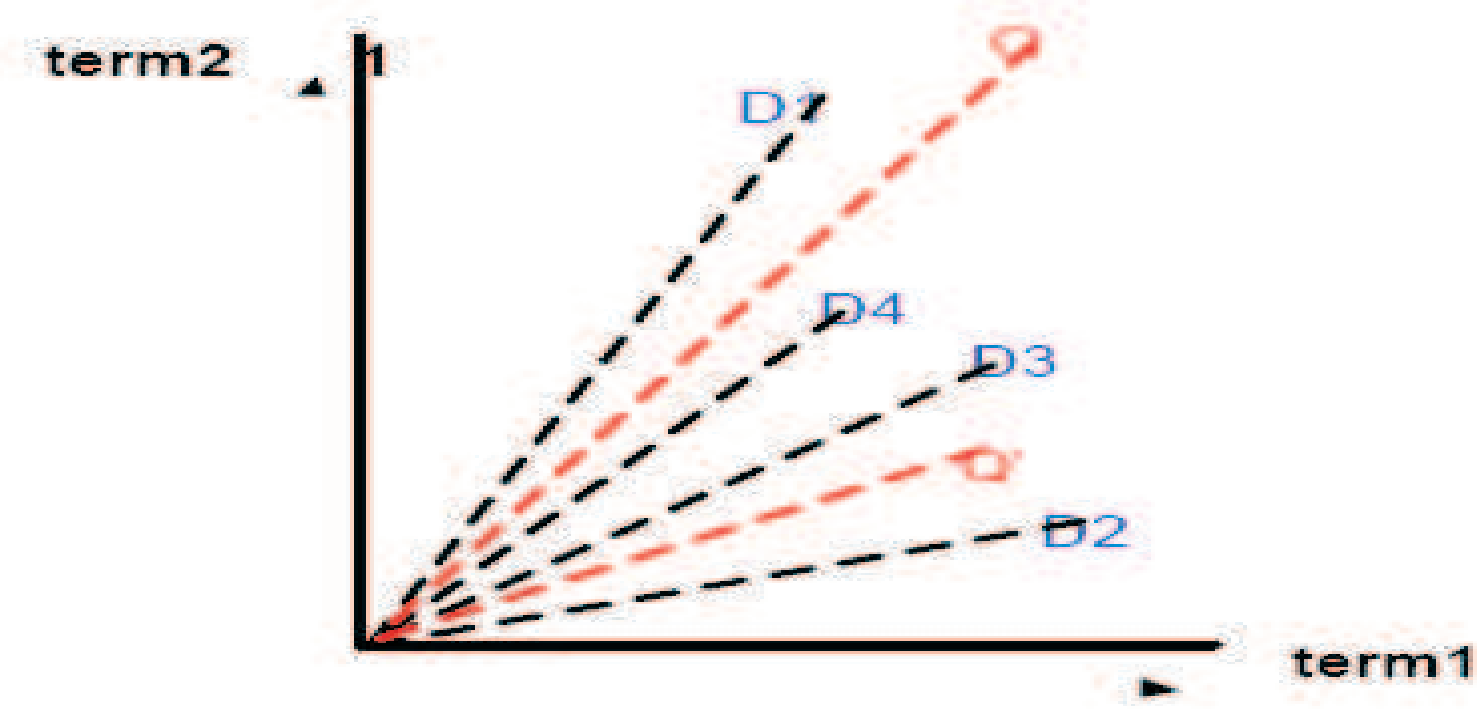
- Añadir términos útiles a la consulta
- Aquéllos que aparecen frecuentemente en los docs relevantes
- Se intenta compensar el efecto de consultas pobres (cortas, ambigüas, con demasiados términos genéricos -poco discriminativos-). Se añaden mejores términos
- Se intenta mejorar el *recall* (exhaustividad). Traer más docs relevantes a posiciones altas del ranking.

# Pesado de términos

- Inicialmente los términos se pesan en función de la consulta inicial
  - Ej.- "patito feo" patito 1 feo 1
  - Ej.- "feo patito feo" patito 1 feo 2
- Pesos usualmente de esquemas populares como tf/idf
- Se asignan nuevos pesos de acuerdo a la importancia en los docs relevantes

# Modelo vectorial

- D2, D3 rels
- D1, D4 no rels
- Acercar Q a D2, D3 y alejarlo de D1,D4



# Por ejemplo, en el modelo vectorial

- D1 = <0.8, 0.1, 0.3, 0, 0, 0.1>
- D2 = <0.2, 0.2, 0.7, 0.8,0.8,0>
- Q = <0.4, 0, 0.8, 0,0,0>
- Si D1 es relevante y D2 no relevante:
  - Tras la expansión
  - Q'= <0.4, 0, 0.8, 0,0,0.1>
  - Tras term reweighting
  - Q"= <0.6, 0, 0.5, 0,0,0.05>

# Formalmente

- Fórmula de Rocchio
- Nuevo vector de la consulta = Viejo vector de la consulta + media de los docs rels - media de los docs no relevantes
- $Q_{nueva} = Q_{ini} + 1/n_1 \sum_{i=1}^{n_1} R_i - 1/n_2 \sum_{i=1}^{n_2} S_i$
- $n_1$ num. docs relevantes recuperados, $n_2$ num. docs irrelevantes recuperados
- $R_i$ vector del doc relevante i recuperado
- $S_i$ vector del doc no relevante i recuperado

# Ejemplo

- $Q_{nueva} = Q_{ini} + 1/n_1 \sum_{i=1}^{n_1} R_i - 1/n_2 \sum_{i=1}^{n_2} S_i$
- Q0 = <0.4, 0, 0.2, 0, 0, 0>
- D1 = <0.8, 0, 0.3, 0.5, 0, 0>
- D2 = <0.1, 0.2, 0.7, 0.1, 0.5, 0>
- D3 = <0.3, 0.3, 0.7, 0.3, 0.2, 0>
- Media rels= <0.4, 0.17, 0.57, 0.3, 0.23, 0>
- D4 = <0, 0.1, 0.7, 0.9, 0.2, 0>
- D5 = <0, 0, 0.8, 0.9, 0.1, 0>
- Media no rels= <0, 0.05, 0.75, 0.9, 0.15, 0>
- Q1 = <0.8, 0.12, 0.02, -0.6, 0.08, 0>

# RF en el modelo vectorial

- Expansión y reweighting en un sólo paso
- Rocchio tiene muchas variantes:
  $Q_{nueva} = \alpha \times Q_{ini} + \beta/n_1 \sum_{i=1}^{n_1} R_i - \gamma/n_2 \sum_{i=1}^{n_2} S_i$
- No es muy formal (a diferencia de otros modelos de RI)

# Modelo Probabilístico

- Alternativa al VSP
- W. Maron, S. Robertson, K. Sparck-Jones, C.J. van Rijsbergen y otros
- Diseñado específicamente para RF
- Muy buena revisión de los MP: Sparck Jones, Walker y Roberton. A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management 36, 2000.

# Modelo Probabilístico

- Se basa en estimar la probabilidad de que un doc esté en la clase de docs relevantes de la consulta
- P(R|D)
- Para estimar esto se suele tomar alguna suposición/simplificación
- p.e. que los términos son independientes entre sí y que la importancia de los términos en los docs se modela como una noción binaria (BIM, Binary Independence Model)

# Modelo Probabilístico

- Los docs se ordenan por el valor $\frac{P(R|D)}{P(\overline{R}|D)}$
- Aplicando la regla de bayes:

  $$P(R|D) = \frac{P(D|R)P(R)}{P(D)}$$

  $$P(\overline{R}|D) = \frac{P(D|\overline{R})P(\overline{R})}{P(D)}$$

  $$\frac{P(R|D)}{P(\overline{R}|D)} = \frac{P(R)}{P(\overline{R})}\frac{P(D|R)}{P(D|\overline{R})}$$

  A efectos de construir un ranking de docs podemos quedarnos con $\frac{P(D|R)}{P(D|\overline{R})}$
- Falta por estimar $P(D|R)$ y $P(D|\overline{R})$

# Modelo Probabilístico

- Falta por estimar $P(D|R)$ y $P(D|\overline{R})$
- (suposición de independencia de los términos) $P(D|R) = \Pi_i P(t_i|R)$
- (términos binarios) $P(t_i|R) = P(t_i = 1|R)$
- Es decir, en el MP es básico estimar qué probabilidad tienen los términos de figurar en los docs relevantes
- Dicho de otro modo, qué prob. hay de que un término de la consulta figure en un documento relevante
- La inf. de feedback es básica en el PM. Los términos de la consulta se van repesando. Los $P(t_i|R)$ van variando a medida de que se disponga de información de relevancia.

# Fórmula F4

- Cuán importante es el término *algebra*?
  - supongamos que la colección contiene 100 docs ($N$)
  - supongamos que el usuario selecciona 5 docs relevantes ($R$)
  - supongamos que 4 docs de la colección contienen el término *algebra* ($n_{algebra}$)
  - supongamos que de los 5 docs rels 4 contienen *algebra* ($r_{algebra}$)
- importancia de *algebra* en los docs rels: $\frac{r_{algebra}}{R-r_{algebra}}$
- importancia de *algebra* en los docs no rels: $\frac{n_{algebra}-r_{algebra}}{N-n_{algebra}-R+r_{algebra}}$

# Fórmula F4

- $\frac{r_{algebra}}{R-r_{algebra}}$ mide
  - rels conteniendo *algebra* frente a rels no conteniendo *algebra*
  - valor alto si la mayoría de los rels tienen el término
  - valor bajo si la mayoría de los rels no tienen el término
- $\frac{n_{algebra}-r_{algebra}}{N-n_{algebra}-R+r_{algebra}}$ mide
  - no rels conteniendo *algebra* frente a no rels que no mencionan *algebra*
  - valor alto si la mayoría de los docs que tienen el término no son rels.
  - valor bajo si la mayoría de los docs que no tienen el término son irrels.

# Fórmula F4

- Peso final:

$$w_{q_i} = log \frac{\frac{r_{algebra}}{R - r_{algebra}}}{\frac{n_{algebra} - r_{algebra}}{N - n_{algebra} - R + r_{algebra}}}$$

- Los documentos son pesados a través de la suma de los términos de la consulta:

$$sim(d_j, q) = \sum_{i=1}^{n} w_{q_i}$$

# Modelo probabilístico

- Esta medida puede usarse también para seleccionar términos para expansión de consultas:
  - Los términos de los docs rels se ordenar por el valor de F4
  - Esto modela cuán buenos son los términos para recuperar docs rels.
  - Añadir todos los términos o sólo unos pocos (p.e. los 5 con mayor valor de F4)
- La expansión y el reweighting se hacen independientemente

# Modelo probabilístico

- Ventajas con respecto al modelo vectorial:
  - Fuerte base teórica (Teoría de la Probabilidad, fácil de extender)
- Desventajas:
  - Los modelos son usualmente más complicados
  - No tf (frecuencia de término en doc)
- Cuál es mejor? rendimiento similar

# RF en sistemas booleanos

- Alterar conectores booleanos:
  - (gato OR perro) $\rightarrow$ (gato AND perro)
  - Difícil para consultas complejas
- Alterar el conjunto de términos:
  - (gato OR perro) $\rightarrow$ (gato OR perro OR mamut)
  - Los términos se seleccionan por medidas como la de F4
  - Se suelen incroporar mediante ORs

# Otras alternativas

- Expansión a través de tesauros
- “Daffy Duck”
- Entry: daffy
- Function: sdjective
- Definition: stupid
- Synonyms: absurd, asisine, baked, bedlamite, ..
- Antonyms: rational, sane, sensible, sound, wise
- Concept: insanity

# Otras alternativas

- los usuarios seleccionan sus propios términos para la expansión
- p.e. a partir de un ranking de términos (F4)
- expansión de consultas *interactiva*

# En resumen

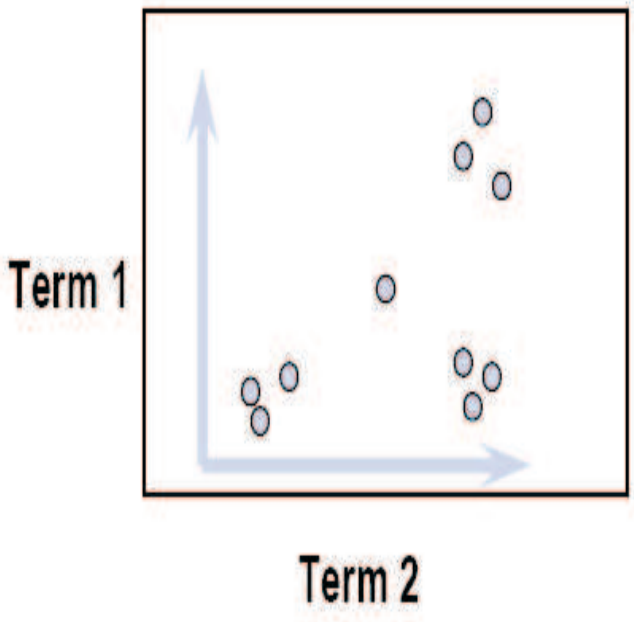
- RF implica alterar la consulta de usuario
- Puede ser muy efectivo si los usuarios lo utilizan
- supone romper el proceso de búsqueda en partes. mejorar gradualmente la consulta inicial
- menos énfasis en la consulta, más énfasis en los docs

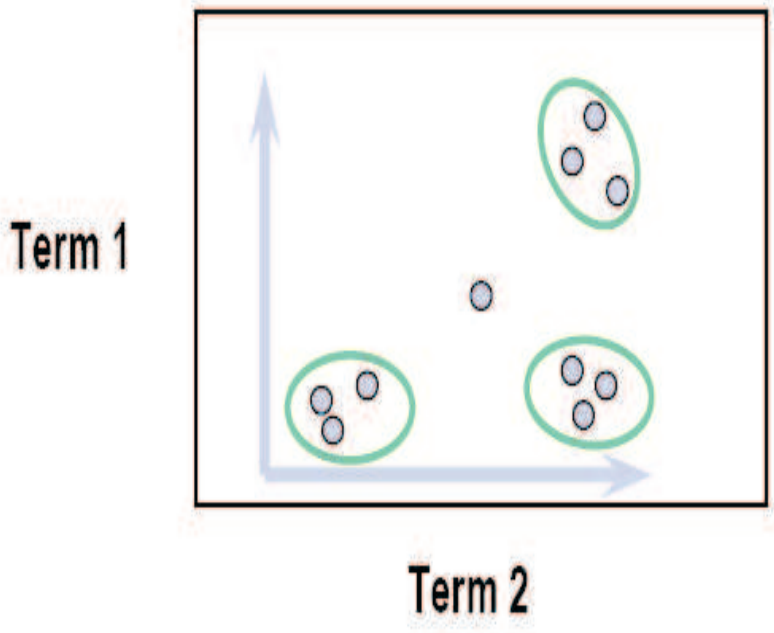
# Clusterización

- El espacio de docs tiene una alta dimensionalidad
- Qué ocurre más allá de 2 o 3 dimensiones?
- Los cálculos de similaridad tienen que ver con cuántos elementos comunes hay
- Más términos $\rightarrow$ más difícil comprender qué subconjuntos de palabras son comunes entre los docs que son realmente similares
- La forma más común de gestionar esta alta dimensionalidad es a través de clusterización
- Otra forma usual es la descomposición por valores singulares (*Singular Value Decomposition*, SVD) utilizada en el indexado por semántica latente (*latent semantic indexing*, LSI)

# Clusterización

- Encontrar grupos en los datos:



# Clusterización

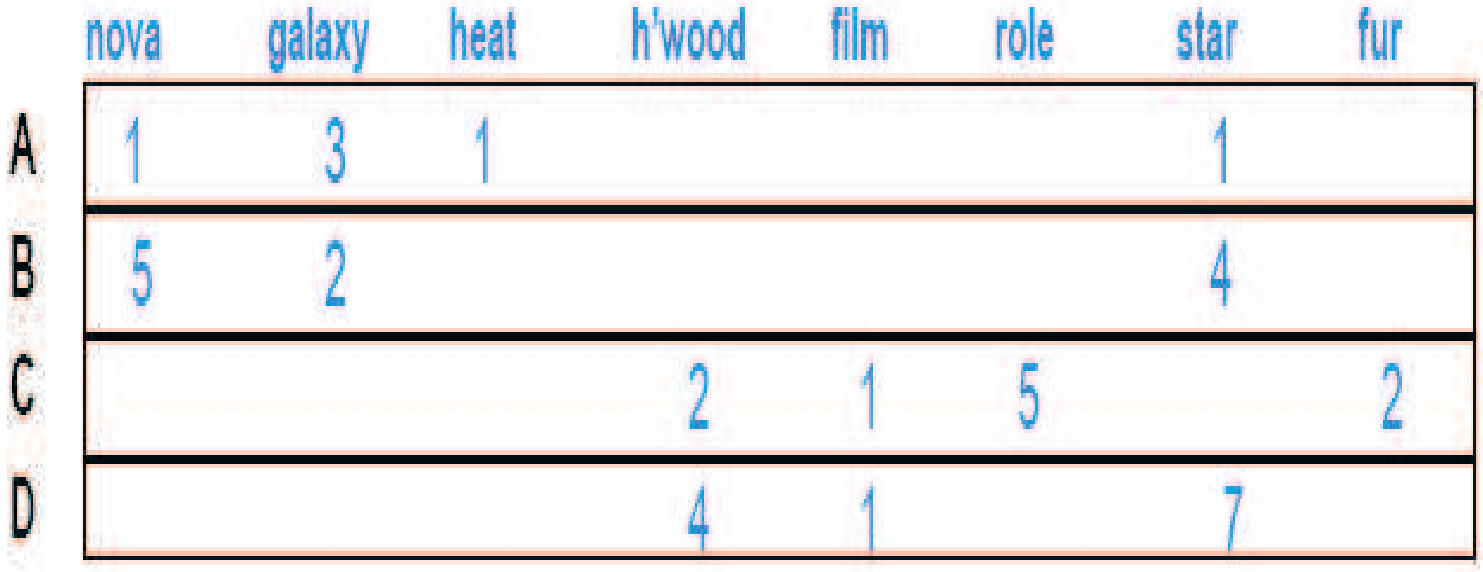
- Agrupar automáticamente docs (o términos) similares en clusters
- Si una colección de documentos está bien clusterizada podemos simplemente buscar en los clusters, en lugar de en la colección entera.
  - Encontrar similaridades entre términos (construir tesauros)
  - Encontrar similaridades entre docs (clasificación)
  - Escoger temáticas e ignorar otras (filtrado)

# Clusterización

- Un tema de investigación muy tradicional en RI
- Inicialmente para almacenar y realizar búsquedas sobre docs más eficientemente
- Ahora se usa para navegar sobre grandes colecciones (e.g. directorio Yahoo)

# Similaridades doc a doc

- Cómo calcular las similaridades?



# Similaridades doc a doc

- Sin normalización

$$D_1 = w_{11}, w_{12}, \ldots, w_{1t}$$

$$D_2 = w_{21}, w_{22}, \ldots, w_{2t}$$

$$sim(D_1, D_2) = \sum_{i=1}^{t} w_{1i} * w_{2i}$$

$$sim(A, B) = (1*5) + (2*3) + (1*4) = 14$$
$$sim(A, C) = 0$$
$$sim(A, D) = (1*7) = 7$$
$$sim(B, C) = 0$$
$$sim(B, D) = (4*7) = 28$$
$$sim(C, D) = (2*4) + (1*1) = 9$$

| | nova | galaxy | heat | h'wood | film | role | star | fur |
|---|---|---|---|---|---|---|---|---|
| A | 1 | 3 | 1 | | | | 1 | |
| B | 5 | 2 | | | | | 4 | |
| C | | | | 2 | 1 | 5 | | |
| D | | | | 4 | 1 | | 7 | |

# Similaridades doc a doc

- Normalización del coseno

$$D_1 = w_{11}, w_{12}, \ldots, w_{1t}$$

$$D_2 = w_{21}, w_{22}, \ldots, w_{2t}$$

$$sim(D_1, D_2) = \sum_{i=1}^{t} w_{1i} * w_{2i} \quad \text{unnormalized}$$

$$sim(D_1, D_2) = \frac{\sum_{i=1}^{t} w_{1i} * w_{2i}}{\sqrt{\sum_{i=1}^{t} (w_{1i})^2 * \sum_{i=1}^{t} (w_{2i})^2}} \quad \text{cosine normalized}$$

# Matriz doc a doc

| $Sim$ | $D_1$ | $D_2$ | $\ldots$ | $D_t$ |
|---|---|---|---|---|
| $D_1$ | . | $s_{12}$ | $\ldots$ | $s_{1t}$ |
| $D_2$ | $s_{21}$ | . | $\ldots$ | $s_{2t}$ |
| . | . | . | . | . |
| . | . | . | . | . |
| $D_n$ | $s_{n1}$ | $s_{n2}$ | $\ldots$ | . |

$sim(D_i, D_j) =$ similaridad de $D_i$ a $D_j$

# Clusterización jerárquica por aglomeració

1. Crear una matriz de sim. doc a doc N x N
2. Cada doc es inicialmente un cluster de tamaño 1
3. Hacer hasta que haya un sólo cluster
   - Combinar los 2 clusters con mayor sim
   - Actualizar la matriz doc a doc

# Clusterización jerárquica por aglomeració

- Varias formas de calcular los rdos de sim producen distintos algoritmos:
  - Enlace simple
  - Enlazado completo
  - Media agrupada
  - Método de Ward

# Ejemplo

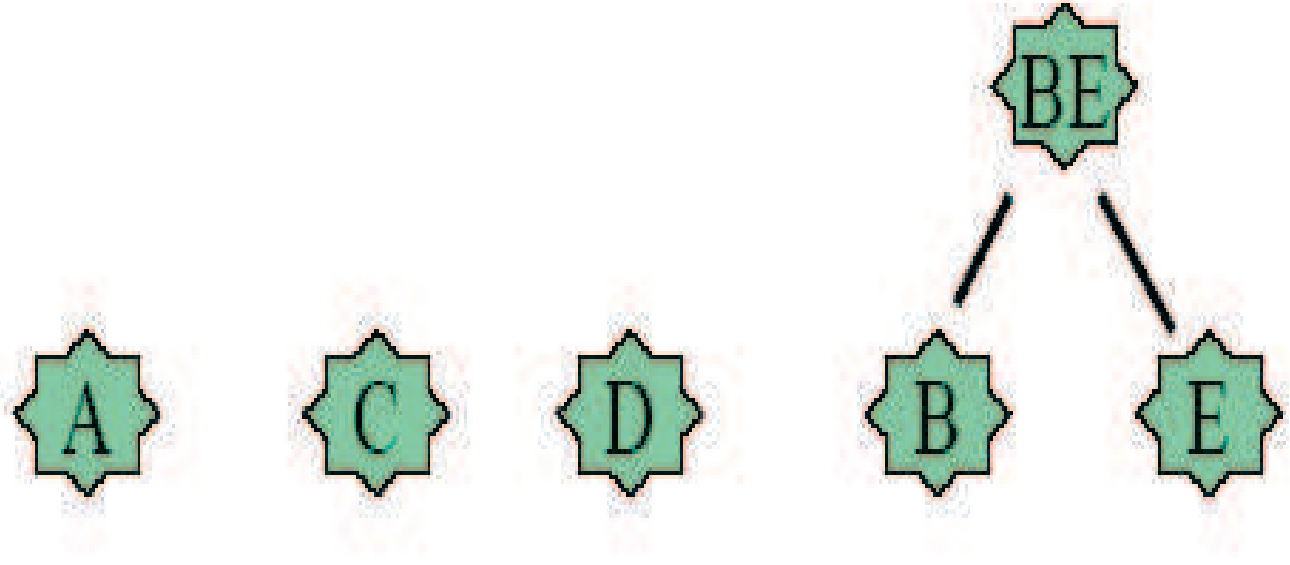
- Sean los 5 docs y similaridades siguientes:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 2 | 7 | 9 | 4 |
| B | 2 | - | 9 | 11 | 14 |
| C | 7 | 9 | - | 4 | 8 |
| D | 9 | 11 | 4 | - | 2 |
| E | 4 | 14 | 8 | 2 | - |

- La sim más alta es sim(E,B)=14

# Ejemplo

- E y B al mismo cluster



# Ejemplo

- Actualización de la matriz doc-doc

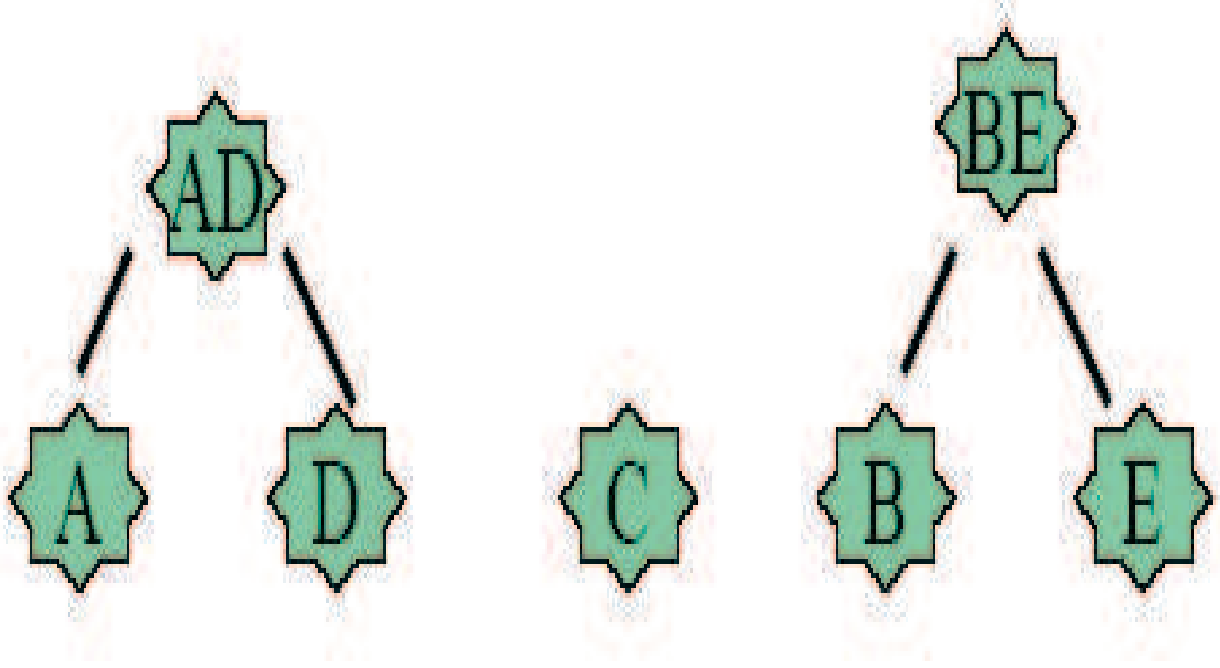
| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 2 | 7 | 9 | 4 |
| B | 2 | - | 9 | 11 | 14 |
| C | 7 | 9 | - | 4 | 8 |
| D | 9 | 11 | 4 | - | 2 |
| E | 4 | 14 | 8 | 2 | - |

| | A | BE | C | D | |
|---|---|---|---|---|---|
| A | - | 2 | 7 | 9 | |
| BE | 2 | - | 8 | 2 | |
| C | 7 | 8 | - | 4 | |
| D | 9 | 2 | 4 | - | |
| | | | | | |

- Para calcular sim(x,BE):
  - $sim(A, BE) = 4$. Enlace simple (máxima sim).
  - $sim(A, BE) = 2$. Enlazado completo (mínima sim). Usaremos esta.
  - $sim(A, BE) = 3$. Media agrupada (sim media).
- ahora sim(A,D) es la máxima similaridad

# Ejemplo

- A y D al mismo cluster



# Ejemplo

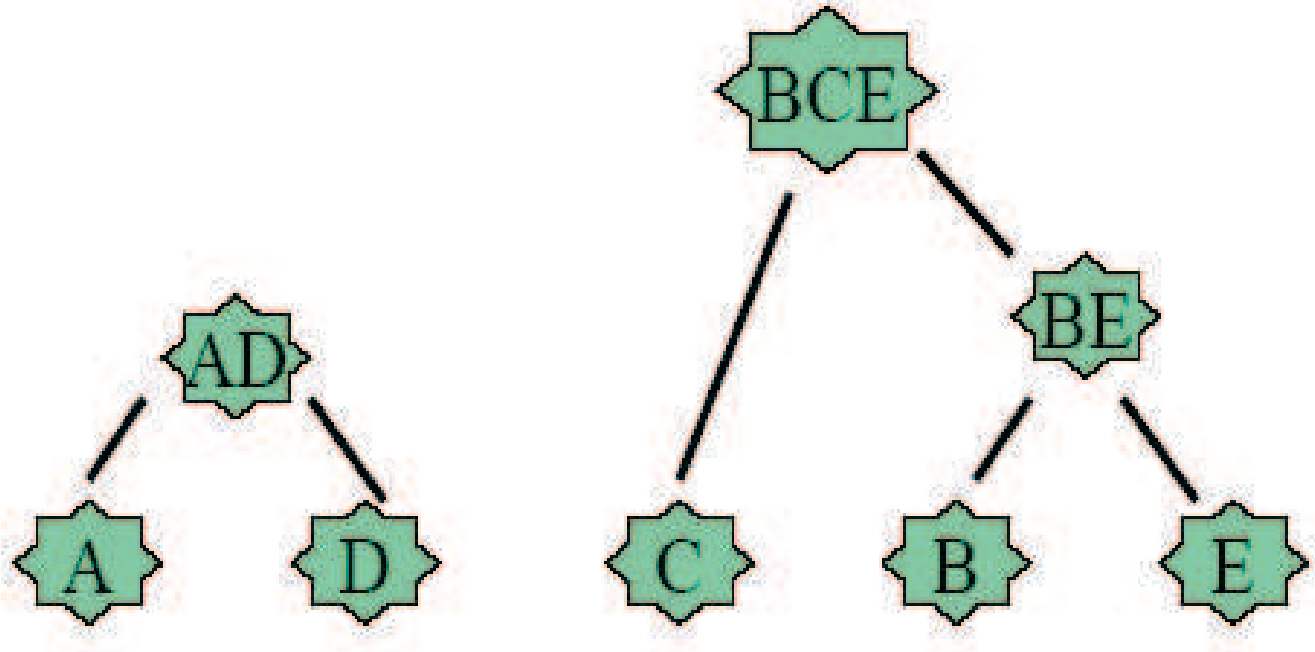
- Actualización de la matriz doc-doc

| | A | BE | C | D | |
|---|---|---|---|---|---|
| A | - | 2 | 7 | 9 | |
| BE | 2 | - | 8 | 2 | |
| C | 7 | 8 | - | 4 | |
| D | 9 | 2 | 4 | - | |
| | | | | | |

| | AD | BE | C | | |
|---|---|---|---|---|---|
| AD | - | 2 | 4 | | |
| BE | 2 | - | 8 | | |
| C | 4 | 8 | - | | |
| | | | | | |
| | | | | | |

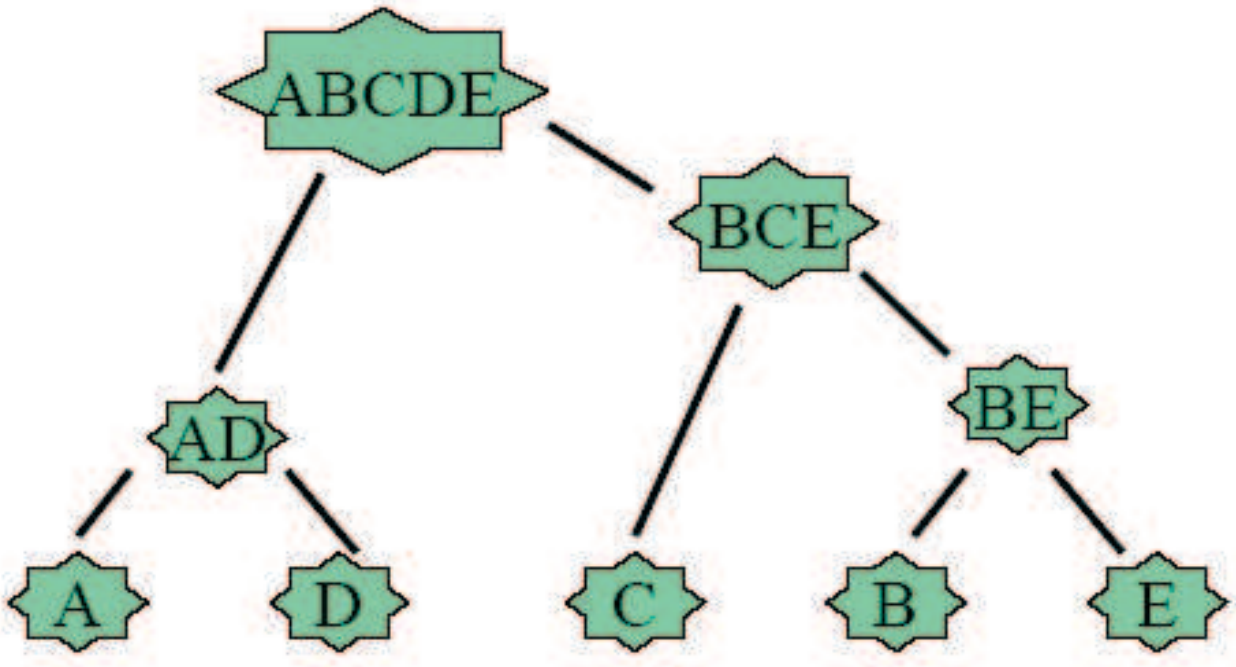
- sim(x,AD). Enlazado completo.
- sim(BE,C) es la max sim.

# Ejemplo



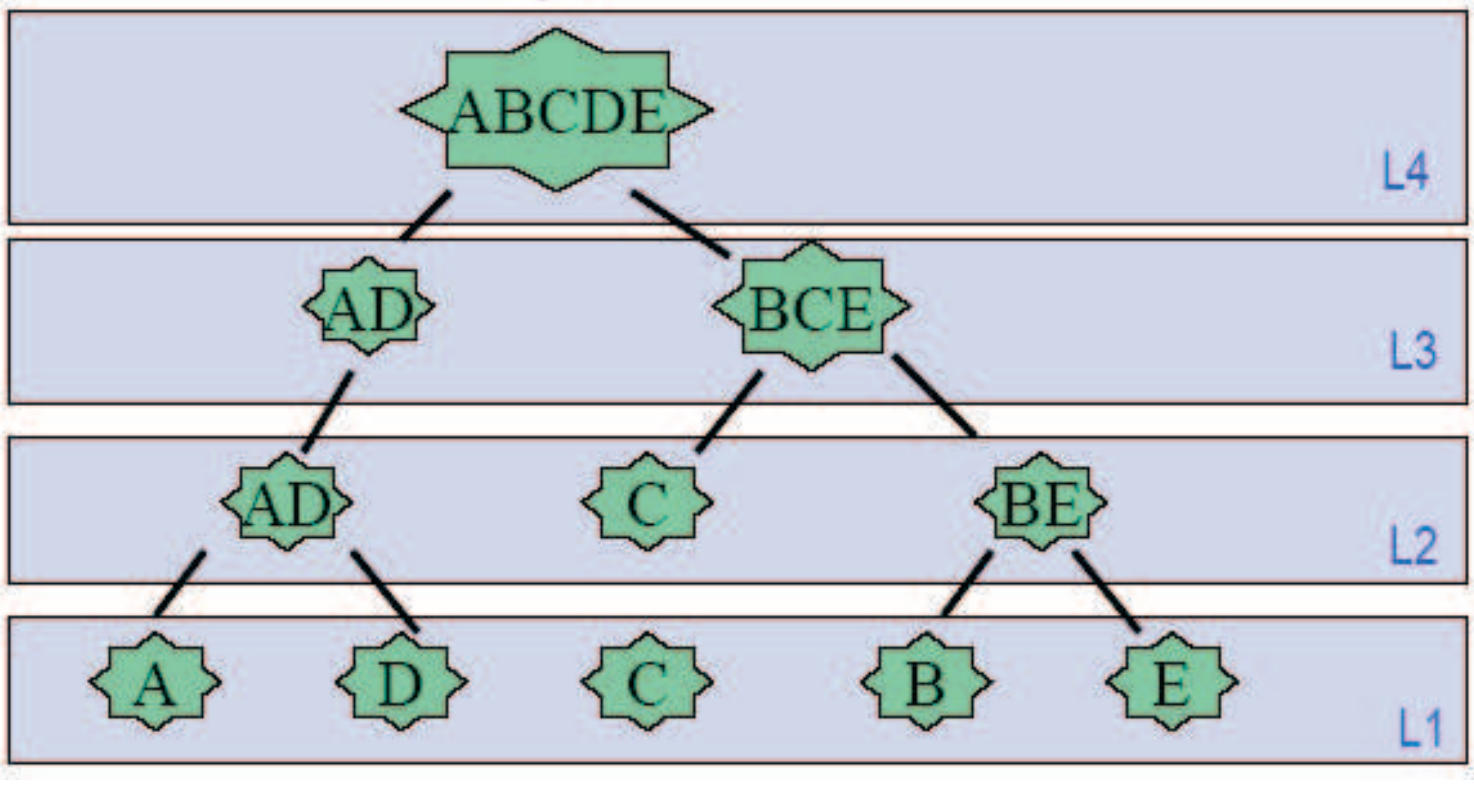
# Ejemplo

- Quedan sólo dos clusters. Los ponemos juntos.



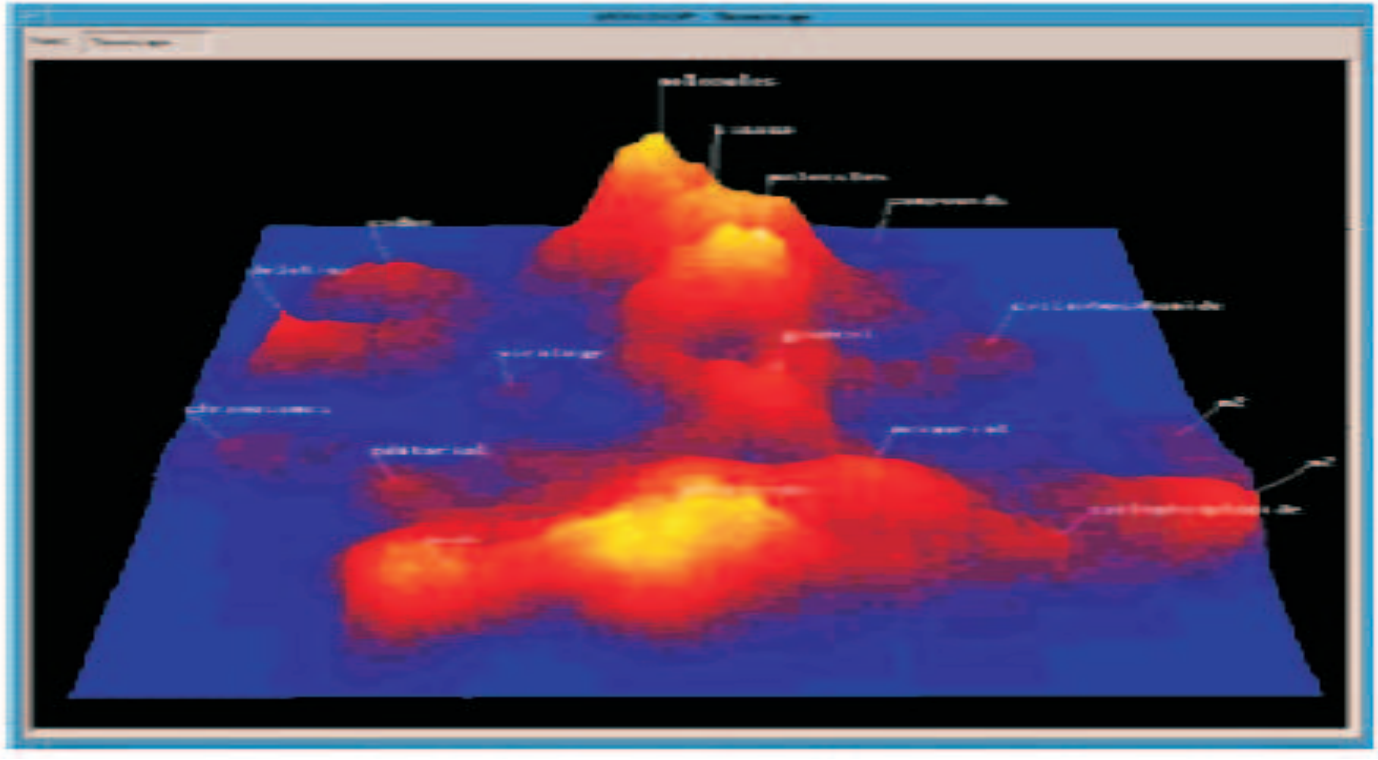
# Ejemplo

- Visualización de agrupamientos a distintos niveles de sim



# Ejemplo

- Visualización en un espacio multidimensional



# Análisis

- El clustering jerárquico es costoso
- $\mathcal{O}\ (n^2)$ para calcular la matriz doc-doc
- Se añade un nodo en casa paso de la clusterización $\mathcal{O}\ (n)$
- Tras añadir el nodo hay que recalcular la matriz para el nuevo cluster, $\mathcal{O}\ (n)$
- Por tanto, $\mathcal{O}\ (n^2)$
- Por ejemplo, 500.000 docs $\rightarrow$ 250.000.000.000 pasos

# Otros algoritmos

- Una pasada
- Buckshot
- Mejores que clusterización jerárquica?

# Clusterización en una pasada

1. Escoger un doc inicial para formar un cluster de tamaño 1
2. Calcular sim con todos los restantes nodos
3. Añadir el más similar al cluster
4. Si no hay ninguno similar (valor umbral) se crea un nuevo cluster entre los dos nodos más similares

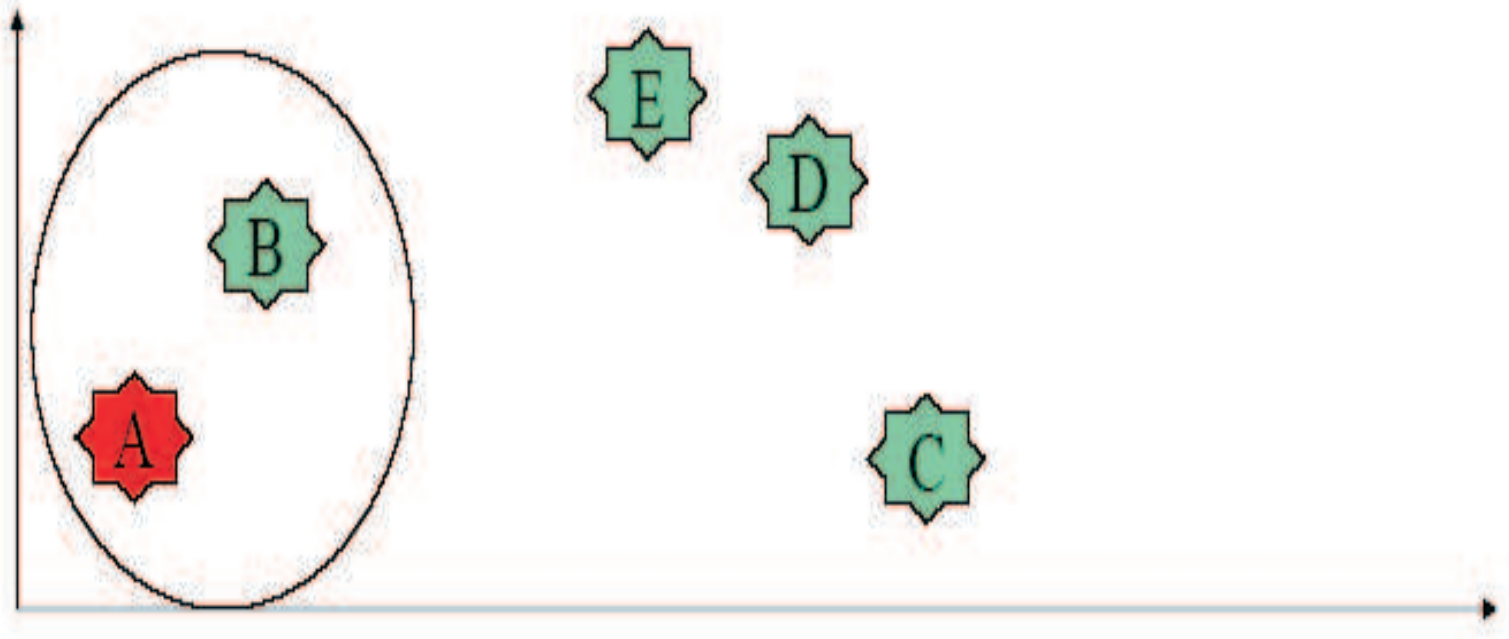
# Ejemplo

- Matriz doc-doc

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 2 | 7 | 9 | 4 |
| B | 2 | - | 9 | 11 | 14 |
| C | 7 | 9 | - | 4 | 8 |
| D | 9 | 11 | 4 | - | 2 |
| E | 4 | 14 | 8 | 2 | - |

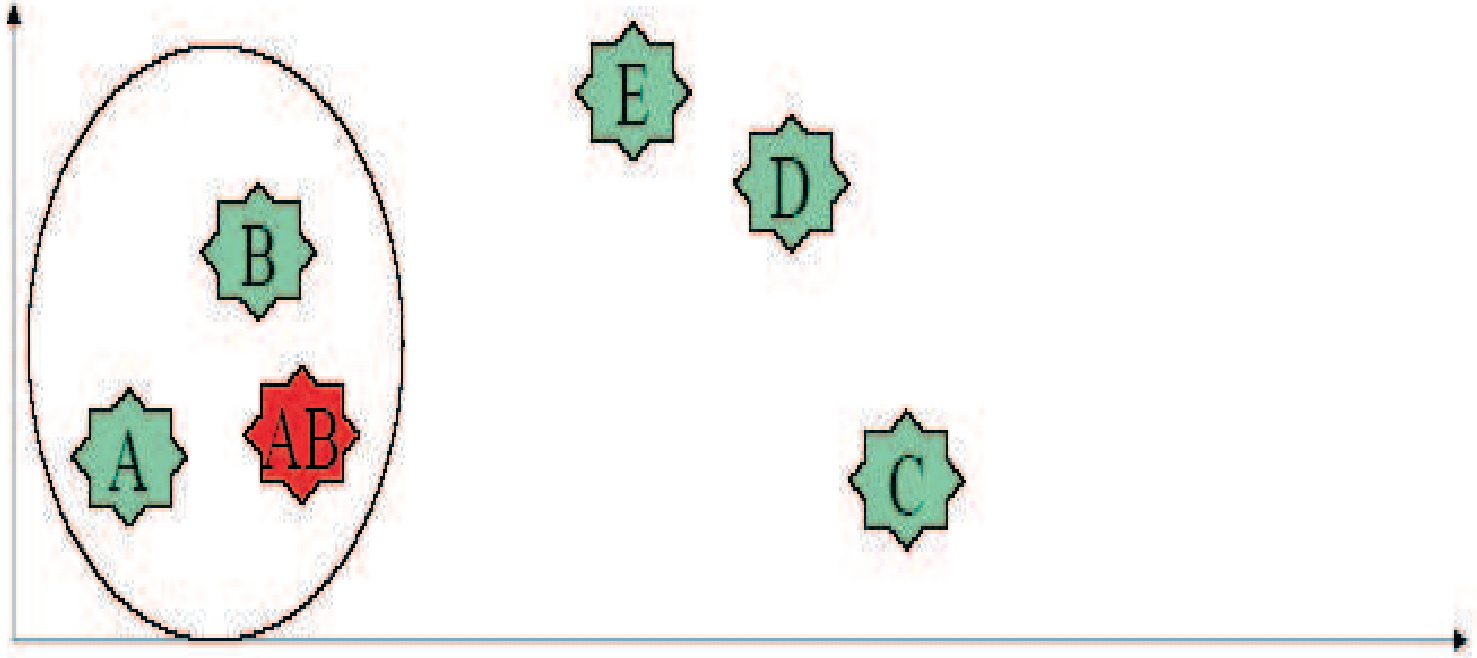
- Tomamos A como el cluster inicial
- El más similar a A es B

# Ejemplo



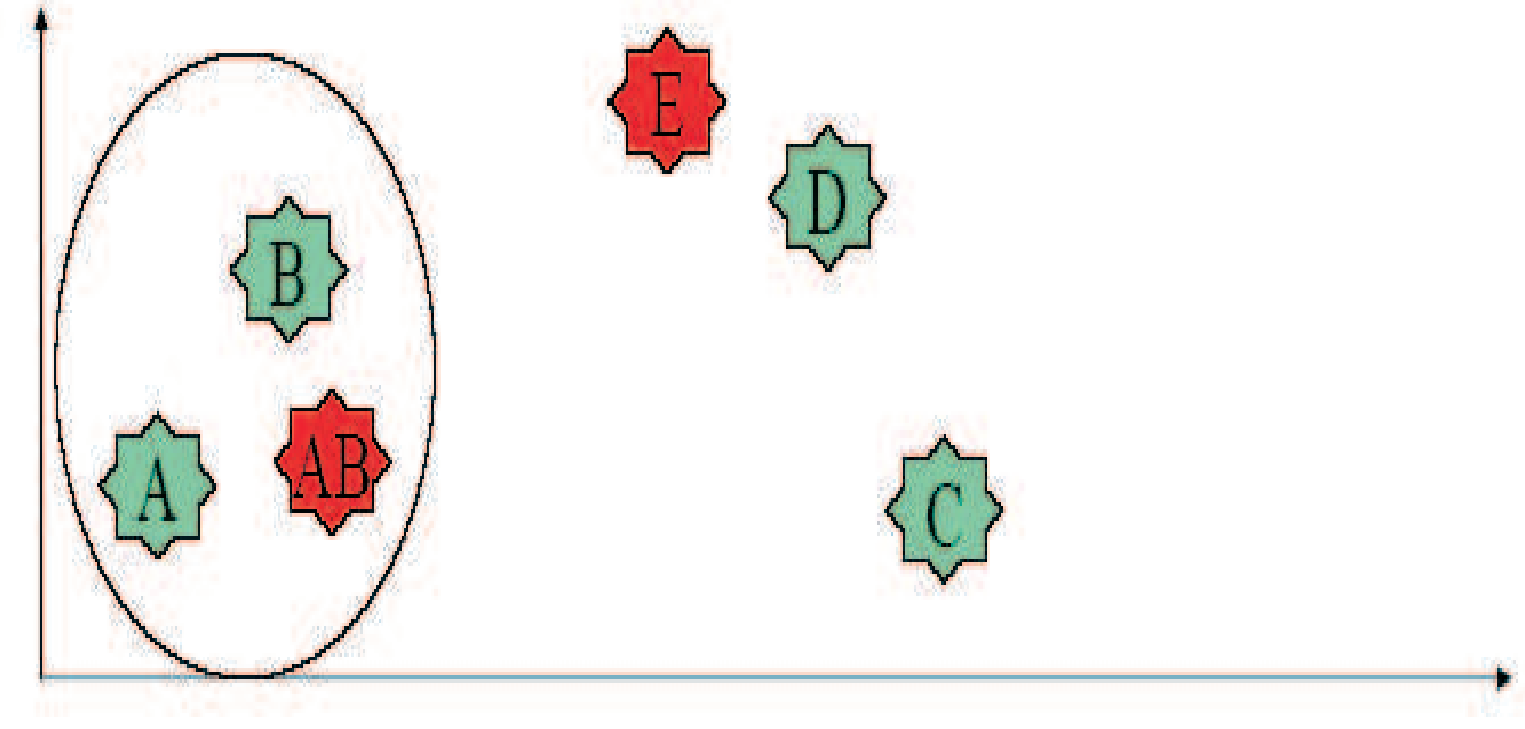
# Ejemplo

- Recalculamos el centroide del cluster



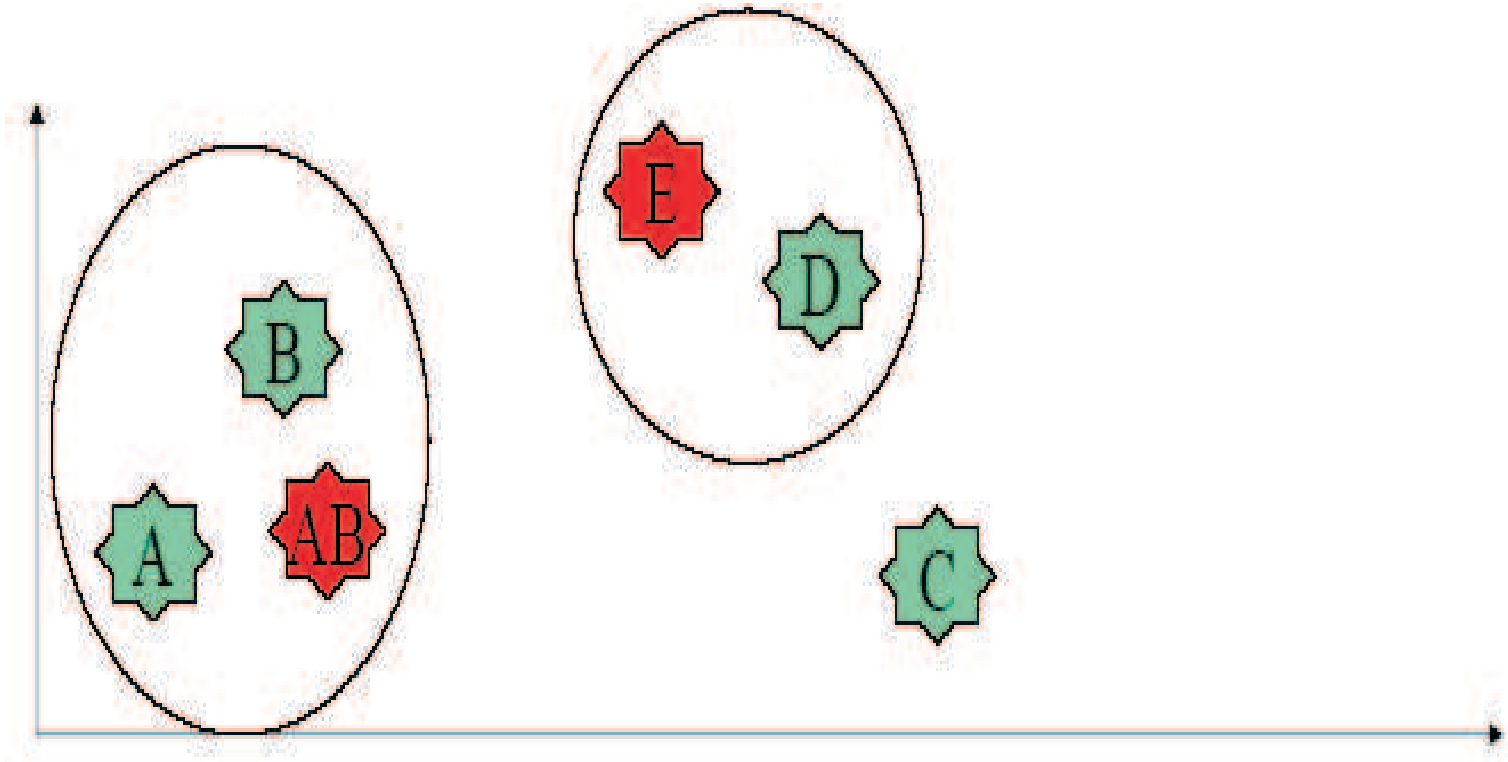
# Ejemplo

- Supongamos que E, D y C son todos muy poco similares a AB. Creamos nuevo cluster, E.



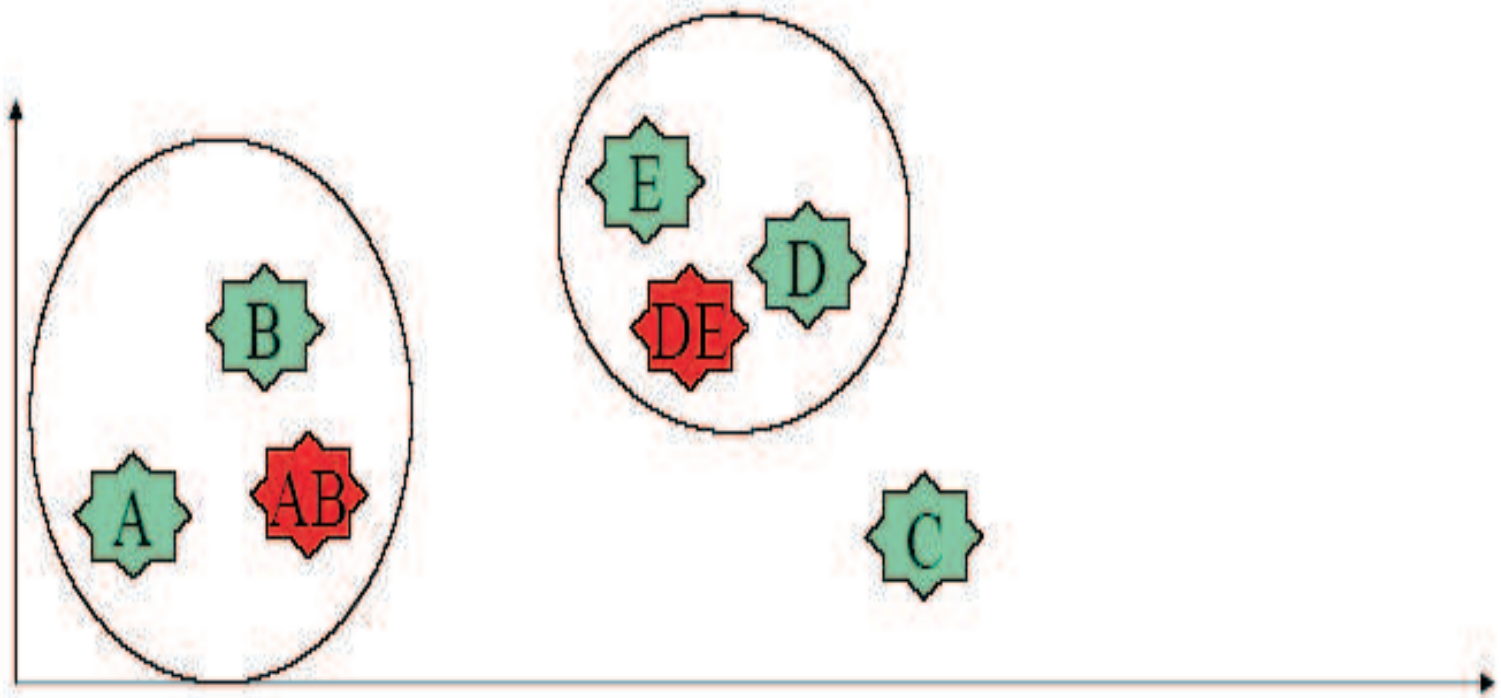
# Ejemplo

- Supongamos que D es más similar a E que C



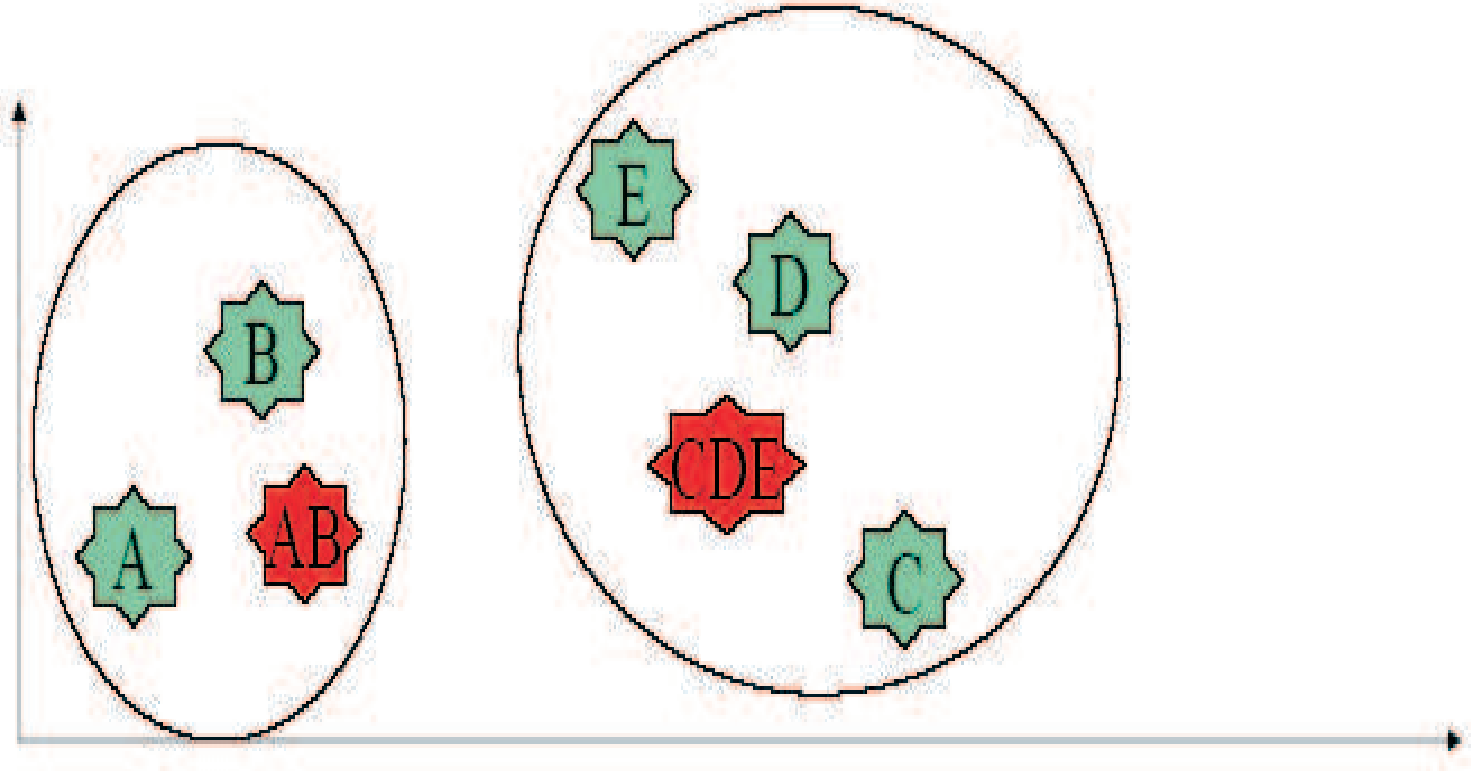
# Ejemplo

- Centroide DE

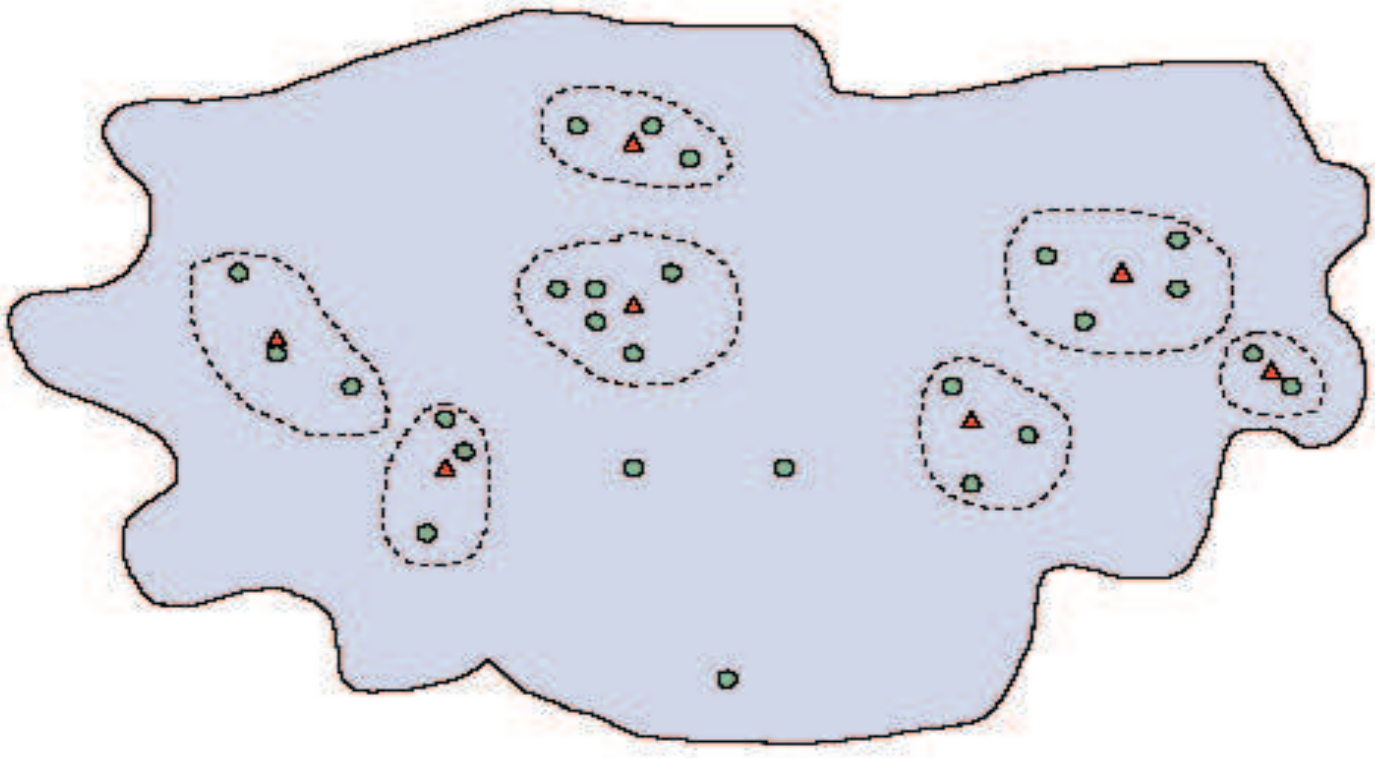


# Ejemplo

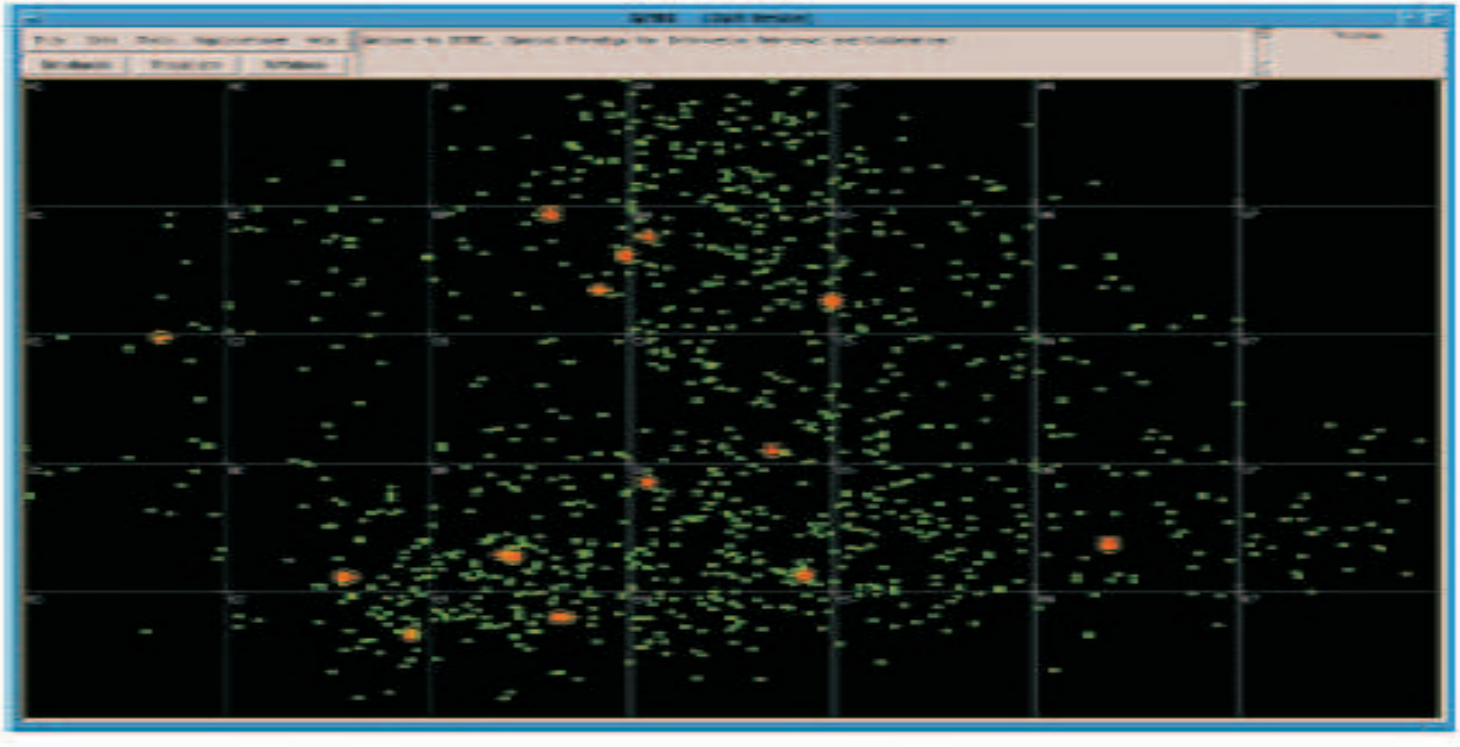
- Calculamos sim(DE,C) si es suficientemente alta:



# Visualización



# Visualización en espacio multidimensiona



# Análisis

- N pasadas (añadimos un nodo en cada pasada)
- La primera pasada require n-1 cálculos de sim
- La segunda requiere n-2
- La última sólo requiere 1
- $1 + 2 + \ldots + n = \frac{n(n-1)}{2}$
- Todavía estamos en $\mathcal{O}\left(n^2\right)$

# Efecto del orden en los docs

- Clusterización en 1 pasada depende de qué doc escogemos como inicial
- Diferentes clusters a partir de distintos docs de partida
- Clusterización jerárquica siempre obtiene los mismos clusters

# Buckshot clustering

- Técnica bastante reciente (1992)
- Obj: pasar de $\mathcal{O}\ (n^2)$ a $\mathcal{O}\ (kn)$, k: num. clusters
  1. Seleccionar aleatoriamente unos docs
  2. Clusterizarlos usando clusterización jerárquica. kn clusters.
  3. Calcular centroides
  4. Asignar los restantes documentos al cluster más cercano
- $\mathcal{O}\ (kn)$+ $\mathcal{O}\ (kn)$+$\mathcal{O}\ (kn)$
- Si k pequeño cerca de orden lineal

# Descomposición por valores singulares

- Es realmente nec. tener tantas dimensiones como términos?
  - Quitar stopwords y stemming no suficiente.
  - Qué pasa con la sinonimia?
  - Tratar con conceptos en lugar de con términos

# Descomposición por valores singulares

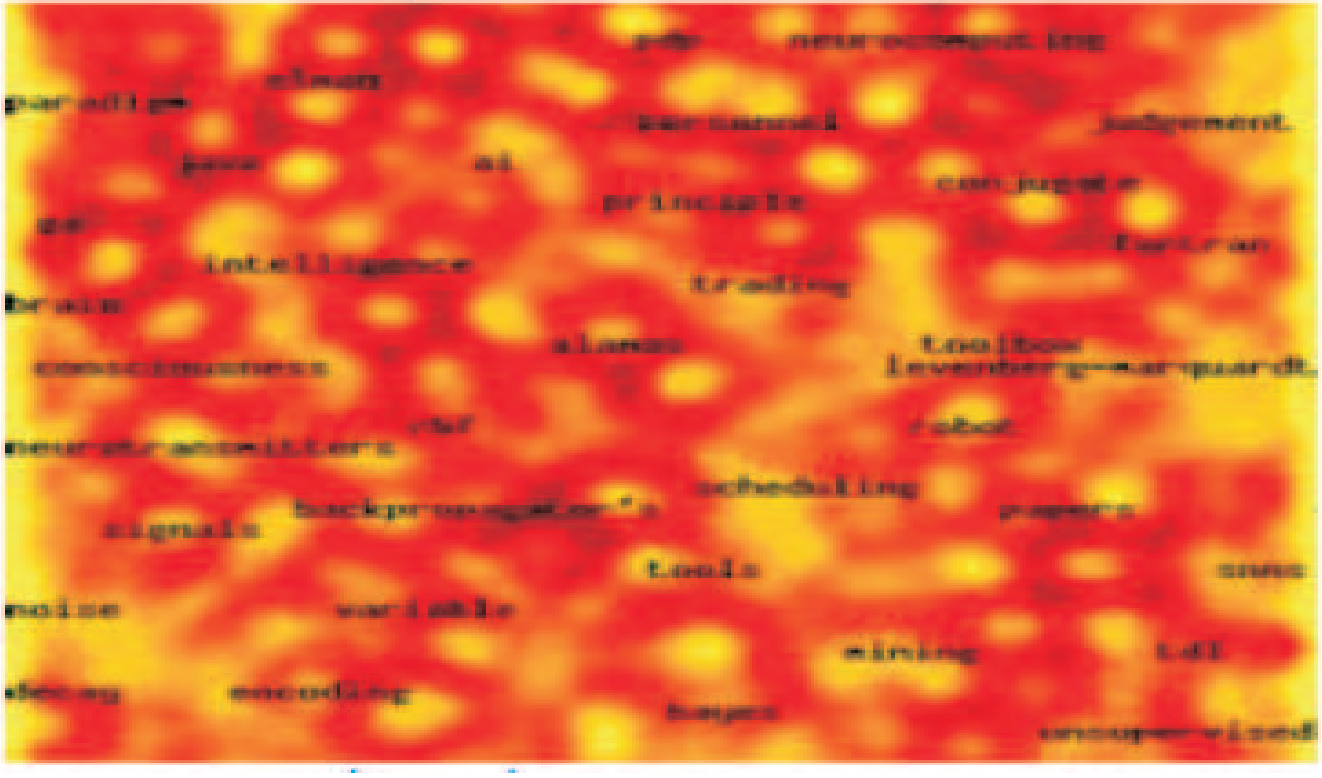
- Encontrar los conceptos *escondidos* bajo el espacio multidimensional (semántica latente)
- Usar una dimensión por concepto
- Descomponer el espacio en “valores singulares”
- Identificar la semántica latente e indexar los documentos por ella (LSI)
- Las consultas se trasladan al espacio de conceptos donde se evalúa su similaridad con los docs

# Clusterización para visualización

- Llevar un enorme espacio multidimensional a un número pequeño de clusters
- Representar estos clusters en 2D
- Identificar temática de cada cluster (no trivial)
- Etiquetar los clusters por temáticas

# Horizontes de conceptos

- Demasiados conceptos o demasiado generalista
- Sólo una temática por doc
- Navegar sin poder hacer búsquedas



# Clusterización

- Ventajas:
  - Funciona para darle al usuario una visión general de los contenidos de una colección
  - Sirve para reducir el espacio de búsqueda
- Desventajas:
  - No hay una técnica de clusterización que sirva para todo
  - Coste computacional
  - Difícil identificar los clusters sobre los que se debe hacer la búsqueda

# CLIR

Cross Language Information Retrieval (CLIR)

- Docs disponibles en muy distintas lenguas
- Recientes estadísticas: sólo el 30% de los usuarios de internet tienen al inglés como lengua madre.
- Creciente interés en manejar situaciones de recuperación en las que hay varias lenguas involucradas
- Tópico de gran actividad: muchos trabajos académicos en CLIR y mucho dinero de investigación para hacer CLIR.

# Qué es CLIR

- La consulta está escrita en una lengua distinta de la de los documentos
- Por qué?
  - para una posterior traducción
  - para leerlos directamente (p.e. nos sentimos más cómodos escribiendo la consulta en español pero no nos importa leer el documento en inglés)
  - en general, para poder acceder a una mayor cantidad de información (no estamos limitados a un lenguaje en concreto)

# CLIR: es efectivo?

- La traducción máquina no es sencilla, ni mucho menos.
- Pero Recuperación de Información (RI) es diferente...
- Los docs y consultas se suelen tratar como bolsas de palabras (los sistemas no suelen implementar NLP avanzado pues no suelen mejorar el estado del arte en RI)
- Los sistemas son bastante tolerantes a algún nivel de error

# CLIR: es efectivo?

- Esto es, aunque la traducción automática no sea totalmente efectiva, el proceso de recuperación puede tolerarlo bastante bien (al final, va a haber un proceso de macheado consulta-documento gobernado por esquemas de pesado de la importancia de los términos en el que la imprecisión de una traducción puede superarse)

# CLIR vs Multi-language retrieval

- Al principio eran sinónimos pero ahora no
- CLIR: recuperar docs que estan escritos en lengua distinta de la de la consulta
- Multi-language IR: sistema que gestiona docs en distintas lenguas pero no realiza traducciones
- Mono-language IR: una sóla lengua

# CLIR: Cómo hacerlo

- Problemas para tomar la traducción correcta: ambigüedades, palabras compuestas (p.e. abdominales vs sit-ups)
- Qué traducimos la consulta o los docs?
- Es más sencillo traducir la consulta pero disponemos de menos términos y, por tanto, de menos evidencia (p.e. para eliminar ambigüedades)
- Es más trabajoso traducir los docs pero la traducción puede ser más precisa

# CLIR: traducción correcta

- Ambigüedad: Grande puede aparecer como big, large, huge, massive,... Numerosas traducciones posibles lo que dificulta el proceso posterior de recuperación
- Términos compuestos: "Petit déjeuner" puede significar "little dinner" o "breakfast".

# Diccionarios bilingües

- Aproximación simple.
- Puede gestionar términos compuestos si el diccionario los contiene
- No maneja ambigüedad
- Es la aproximación más común

# Corpus paralelos

- Se dispone de dos traducciones de los mismos textos
- “Le chien et dans le jardin. La chat et sur la table”
- “The dog is in the garden. The cat is on the table”
- Es necesario disponer de un número elevado de textos traducidos a ambas lenguas.
- Cada vez que se quiere traducir algo (p.e. una consulta) se busca en el corpus paralelo
- Puede gestionar términos compuestos y ambigüedad (limitado siempre a que existan ejemplos de todos los términos en el corpus)
- No es una aproximación muy estándar (costoso hacerse con un corpus significativo)

# Corpus comparables

- No son traducciones directas de los mismos textos
- Por ejemplo, extraídos de periódicos (noticias cubriendo los mismos acontecimientos)
- Ventaja: más fácil obtener corpus de este tipo (no hay que traducirlos)
- Puede gestionar términos compuestos y ambigüedad (limitado siempre a que existan ejemplos de todos los términos en el corpus)
- Tampoco es una aproximación muy estándar

# Expansión de consultas

- Expandir la consulta antes de traducirla (p.e. a partir de una colección de datos en el lenguaje de la consulta)
- Expandir tras la traducción

# Más problemas

- Qué pasa si no disponemos de recursos para hacer la traducción?
- P.e. de portugués a griego
- Usar un lenguaje intermedio (lenguaje pivote)
- Portugués $\rightarrow$ Inglés $\rightarrow$ Griego

# Más problemas

- Podemos usar dos pivotes
- Portugués $\rightarrow$ Inglés $\rightarrow$ Griego
- Portugués $\rightarrow$ Francés $\rightarrow$ Griego
- Cruzamos los resultados

# Mezclando rankings

- Consulta en un lenguaje (p.e. inglés) y docs en muchos lenguajes (p.e. francés, español, portugués,...)
- La calidad de traducción puede ser muy distinta
- Cómo conseguir que la distribución sea justa?

# Referencias

- TREC tracks http://trec.nist.gov
- Cross Language Evaluation Forum (CLEF) http://www.clef-campaign.org
- Trabajos de Lisa Ballesteros. CL retrieval via Transitive Translation.
- Lenguajes de consulta sofisticados y expansión de consulta
- "grand avion" pasa a "SYNONYM(big,large,huge,massive), SYNONYM(plane,aeroplane)"
- Trabajo de Ballesteros dio muy buenos resultados de rendimiento