Departamento de
Lenguajes y Computación

Universidad de Almería

Práctica 1
Informática Documental

Profesor: Antonio Becerra Terón

Aula de Prácticas: Aula 4, CITE III

Octubre, 2007

PRÁCTICA 1. Implementación de un sistema de recuperación espacio-vectorial I. Ponderación de términos e indización de documentos — 9 horas

Objetivo general de la práctica

OBJETIVO

El objetivo básico de esta práctica es abordar la primera fase en la implementación de un sistema de recuperación de información sencillo, basado en el modelo espacio-vectorial. Este objetivo abarca la selección de los términos de indización, ponderación automática, generación de estructuras de indización y, por último, almacenamiento de estas estructuras.

Descripción de la práctica

DESCRIPCIÓN

Esta práctica pretende que el alumno implemente el proceso de selección de términos de indización, ponderación automática e indización, para un sistema de recuperación textual.

La indexación se refiere a la caracterización de documentos para la tarea de la recuperación de información. Qué sería lo ideal? Que un documento indexado sea una representación eficiente de los contenidos semánticos del documento original. Normalmente, la indexación consiste en obtener una lista de *términos* significativos (también denominados *conceptos*) con información asociada (frecuencia en el documento, frecuencia en la base de datos documental, etc). Qué necesitamos ahora? Una técnica que nos permita obtener estas listas de términos.

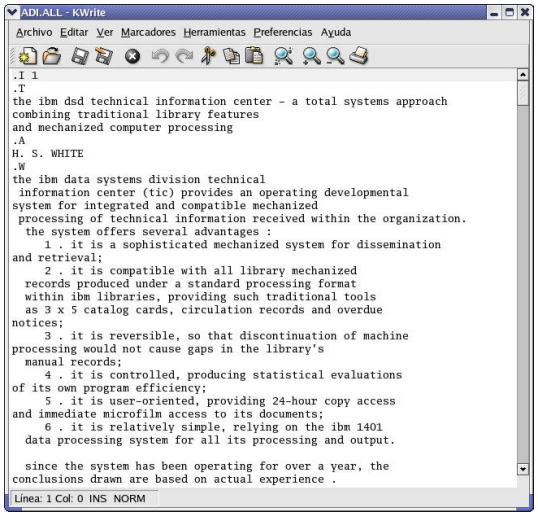
La técnica clásica de recuperación de información supone considerar cada documento como un conjunto de cadenas concatenadas, intentando eliminar las ambiguedades del lenguaje natural que puedan aparecer en los documentos. Las actividades que se realizan en este enfoque de la recuperación de información son:

- Eliminar las palabras que no expresan ningún contenido (*stopwords* o *palabras vacías*);
- Reducir las palabras a algún formato base como puede ser la raíz de cada una de ellas (*stemming* o *lematización*);
- Obtener un peso para los términos de indexación de acuerdo a su relevancia (frecuencia en el documento, frecuencia en la base de datos);

Para el desarrollo de la práctica, en primer lugar el alumno deber diponer de una colección de documentos en formato texto a almacenar en el sistema de recuperación que queremos implementar. Estos documentos proceden de la colección

del grupo de investigación *Information Retrieval* de la Universidad de Glasgow, liderado por el profesor *C. J. van RIJSBERGEN*. La colección la podemos encontrar en la pgina Web de la asignatura, concretamente en la URL http://www.ual.es/∼abecerra/ID/. Esta colección contiene un total de 82 documentos estructurados de la siguiente forma

Figura 1: Documentos en la colección para práctica 1



Aquí, podemos comprobar que I representa el identificador de documento, T el título, A el autor, y W el texto a indexar de forma automática utilizando las técnicas de búsqueda en texto completo.

Una vez entregado el fondo documental, al alumno debe desarrollar un programa que permita realizar una selección automatizada de los términos de indización, y generación de las estructuras de indización. La figura 2 muestra, de forma gráfica, el proceso de esta práctica.

El proceso de indexación de un fondo documental conlleva realizar el *proceso de filtrado* y el *proceso de extracción de los términos relevantes o keywords*. El *proceso de filtrado* supone eliminar todas las cadenas de texto no relevantes y reducir las palabras a un formato base estándar. Este proceso de filtrado está involucrado por los siguientes subprocesos:

(a) Listar las palabras de los documentos

(b) Poner todo el texto en minúscula, y eliminar los símbolos especiales;

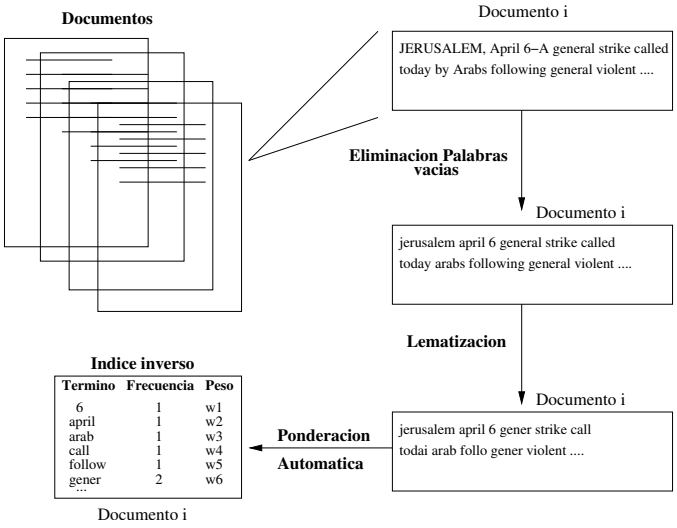
(c) Eliminar de los documentos el conjunto de *palabras vacías* o *stopwords*;

(d) Realizar la reducción de términos por *lematización* o *stemming*;

A continuación, desglosamos las actividades más importantes.

(c) Para la eliminación de las palabras vacías o *stopwords*, el alumno podrá disponer de la lista de palabras vacías propuesta por C. J. van Rijsbergen.

Figura 2: Proceso de la práctica 1 de Informática Documental



Esta lista contiene un conjunto bastante completo de palabras que no tienen contenido semántico para el documento. Entre estas palabras, podemos incluir artcíulos, preposiciones, conjunciones, adverbios, adjetivos, palabras muy cortas, etc.

El alumno debe implementar un programa que lea el texto del conjunto de documentos incluidos en el sistema de recuperación documental, y elimine todas las palabras vacías que aparezcan en dichos documentos. Para la implementación de este programa, se proporciona al alumno un archivo tipo texto, que contiene la lista de palabras vacías que debe comparar. Esta lista de palabras vacías se encuentra en el web de la asignatura, dirección URL http://www.ual.es/∼abecerra/ID/. La implementación del programa deber utilizar una estructura de indexación adecuada para la comparación de las palabras leídas en los documentos con las palabras que aparecen en la lista.

La salida que debe generar el programa, es el conjunto de documentos de la colección, donde se han eliminado todas las palabras vacías, y, por tanto, las palabras que aparecen son susceptibles de considerarse como palabras relevantes o *keywords* para la indización de dicho documento.

(d) El proceso de lematización consiste en reducir un término hasta quedarnos con su raíz, que es, precisamente, la que aporta mayor contenido semántico. Debemos tener en cuenta que el número de palabras derivadas a partir de una misma raíz puede ser muy elevado. Esto supone que indizar documentos por la raíz de los términos proporciona una elevada disminución en el número de términos a considerar.

Para este proceso de lematización o *stemming*, el alumno deber implementar el algoritmo de lematización más utilizado en este proceso dentro del

contexto de la recuperación de información. Se proporciona al alumno este algoritmo en pseudo-cdigo, y el alumno debe aplicarlo a las palabras posiblemente relevantes de los documentos resultantes del proceso de filtrado anterior.

A continuación, mostramos, el pseudo-cdigo del algoritmo que facilitamos al alumno para su implementación:

Consonante: es una letra distinta de A, E, I, O, U, y distinta de Y precedida por una consonante;
Vocal: letra que no es consonante;
Una consonante se denota por c, una vocal por v. Una lista ccc. . . de longitud superior a 0 se denota por C, y una lista vvv. . . por V; tenemos, por tanto, que cualquier palabra o parte de una palabra, tiene uno de los siguientes cuatro formatos:
CVCV . . . C
CVCV . . . V
VCVC . . . C
VCVC . . . V
Esto se puede representar de una única forma como sigue
[C]VCVC . . . [V]
donde [ ] los corchetes indican presencia arbitraria de contenido. Ahora, utilizando (VC){m} para denotar VC repetida m veces, tenemos que esto lo podemos escribir como:
[C](VC){m}[V].
m se denomina la *medida* de la palabra. El caso m=0 representa palabras nulas. Algunos ejemplos seran:
m=0 TR, EE, TREE, Y, BY.
m=1 TROUBLE, OATS, TREES, IVY.
m=2 TROUBLES, PRIVATE, OATEN, ORRERY.
Ahora, las reglas para eliminar un sufijo serán de la forma:
(condición) $S_1 \rightarrow S_2$
Esto significa que si una palabra finaliza con el sufijo S_1, y el lema anterior a S_1 satisface la condición, entonces S_1 es reemplazado por S_2. La condición, normalmente, se proporciona en términos de m; por ejemplo $(m > 1)$ EMENT $\rightarrow$. Permitiría reemplazar REPLACEMENT por REPLAC, dado que REPLAC es una parte de palabra con m=2.
La condición de la regla puede tener los siguientes formatos:
*S - El lema finaliza con S (igual para el resto de letras).
v - El lema contiene una vocal.
*d - El lema finaliza con una consonante doble (i.e. -TT, -SS).
*o - El lema finaliza con cvc, donde la segunda c no es W, X, o Y (i.e. -WIL, -HOP).
Adems, la condición puede incluir expresiones con operadores and, or y not. De forma que,
$(m > 1$ and $(*S$ or $*T))$
comprueba para un lema con $m > 1$, si finaliza en S o T, mientras que
(*d and not (*L or *S or *Z))
comprueba si un lema finaliza con una doble consonante distinta de L, S o Z.
Podemos encontrar regla de reemplazamiento con condiciones nulas; por ejemplo, con
SSES $\rightarrow$ SS
IES $\rightarrow$ I
SS $\rightarrow$ SS
S $\rightarrow$
CARESSES se corresponde con CARESS. Igualmente CARESS se corresponde con CARESS, y, por último, CARES con CARE. A continuación, comenzamos a presentar el algoritmo.

Paso 1a	
SSES $\rightarrow$ SS	caresses $\rightarrow$ caress
IES $\rightarrow$ I	poines $\rightarrow$ poni
	ties $\rightarrow$ ti
SS $\rightarrow$ SS	caress $\rightarrow$ caress
S $\rightarrow$	cats $\rightarrow$ cat

Paso 1b	
(m > 0) EED → EE	feed → feed
	agreed → agree
(*v*) ED →	plastered → plaster
	bled → bled
(*v*) ING →	motoring → motor
	sing → sing

Si la segunda o tercera de las reglas en el Paso 1b tiene éxito, entonces realizar lo siguiente:

AT → ATE	conflat(ed) → conflate
BL → BLE	troubl(ed) → trouble
IZ → IZE	siz(ed) → size
(*d and not (*L or *S or *Z)) → una letra	hopp(ing) → hop
	tann(ed) → tan
	fall(ing) → fall
	hiss(ing) → hiss
	fizz(ed) → fizz
(m = 1 and *o) → E	fail(ing) → fail
	fil(ing) → file

Esta regla de hacer corresponder una única letra provoca la eliminación de uno de los pares de letras dobles.

Paso 1c	
(*v*) Y → I	happy → happi
	sky → sky

Esta regla trata con los plurales y los participios del pasado. Los siguientes pasos del algoritmo son mucho más directos.

Paso 2	
(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	differently → 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 → sentitive
(m > 0) BILITI → BLE	sensibiliti → sensible

Paso 3	
(m > 0) ICATE → IC	triplicate → triplic
(m > 0) ATIVE →	formative → form
(m > 0) ALIZE → AL	formalize → formal
(m > 0) ICITI → IC	electriciti → electric
(m > 0) ICAL → IC	electrical → electric
(m > 0) FUL →	hopeful → hope
(m > 0) NESS →	goodness → good

Paso 4	
(m > 1) AL →	revival → reviv
(m > 1) ANCE →	allowance → allow
(m > 1) ENCE →	inference → infer
(m > 1) ER →	airliner → airlin
(m > 1) IC →	gyroscopic → gyroscop
(m > 1) ABLE →	adjustable → adjust
(m > 1) IBLE →	defensible → defens
(m > 1) ANT →	irritant → irrit
(m > 1) EMENT →	replacement → replac
(m > 1) MENT →	adjustment → adjust
(m > 1) ENT →	depedent → depend
(m > 1 and (*S or *T)) ION →	adoption → adopt
(m > 1) OU →	homologou → homolog
(m > 1) ISM →	communism → commun
(m > 1) ATE →	activate → activ
(m > 1) ITI →	angulariti → angular
(m > 1) OUS →	homologous → homolog
(m > 1) IVE →	effective → effect
(m > 1) IZE →	bowdlerize → bowdler

Paso 5a	
(m > 1) E →	probate → probat
	rate → rate
(m = 1 and not *o) E →	cease → ceas
Paso 5b	
(m > 1) and *d and *L) → letra nica	controll → control
	roll → roll

Una vez implementado el algoritmo, utilizamos el lematizador sobre los documentos para generar el conjunto de palabras relevantes o *keywords* dentro de cada documento. Estas palabras constituyen una representación del texto, tal que los términos se pueden comparar con los términos de otros documentos y los términos que aparecen en las consultas.

La siguiente parte de la práctica se centra en el *proceso de extracción de las keywords*. En este caso, tenemos que los términos proporcionados por el lematizador serán utilizados como términos de indización para los documentos. A cada término de un documento se le asigna un peso de acuerdo a su frecuencia en el documento y en la colección. Antes de introducir lo que debe realizar el alumno en esta última parte de la práctica, definimos los conceptos de frecuencia de un término un documento y frecuencia de documentos de un término.

- Frecuencia de un término i en un documento j: número de ocurrencias del término i en el documento j. Cuantos más aparezca en un documento un término determinado, tenemos que este documento tratará sobre el concepto representado por el término;

- Frecuencia de documentos para un término i: número de documentos indizados por el término i, es decir número de documentos en la colección que contienen el término. Si tenemos términos que aparecen en muchos documentos, entonces estos términos no serán discriminantes para estos documentos.

A continuación, especificamos, de forma más detallada, el proceso de extracción de las palabras relevantes y su ponderación. Este proceso de ponderación debe iniciarse con el cómputo automático de la relevancia de los términos de indización presentes en cada uno de los documentos. La idea es poder generar vectores de documentos

$$d_i \rightarrow \quad \vec{d_i} = (w(t_1, d_i), \ldots, w(t_k, d_i))$$

donde d_i es el documento i, $t_1, \ldots, t_k$ son los k términos de indización obtenidos después de aplicar el lematizador, y $w(t_j, d_i)$ expresa la relevancia del término de indización t_j en el documento d_i.

Por tanto, el alumno debe implementar un programa que le permita calcular, de forma automática, los vectores de ponderación o relevancia de los documentos originales con respecto a los términos de indización. Este programa calculará la ponderación o relevancia de los términos con la fórmula de ponderación que se utiliza habitualmente en el contexto de la recuperación de información.

$$w(t_i, d_j) = w_{i,j} = \frac{F_{i,j} \times Idf_i}{\mid \vec{d_j} \mid} = \frac{F_{i,j} \times \log \frac{N}{n_i}}{\sqrt{\Sigma_{s=1..k}(F_{s,j} \times \log \frac{N}{n_s})^2}}$$

donde $F_{i,j}$ representa la frecuencia del término i en el documento j, Idf_i la inversa de la frecuencia de documentos del término i, $\mid \vec{d_j} \mid$ representa el módulo del vector $\vec{d_j}$, N el número de documentos en la colección, y n_i es la cantidad de documentos donde aparece t_i.

Este mecanismo de cómputo de la ponderación implica recorrer las palabras obtenidas del lematizador para cada documento, y calcular, en función de la frecuencia de aparición de un término lematizado, su importancia o relevancia dentro de un documento. Además, si un término aparece mucho en un documento, se supone importante, pero si aparece en muchos documentos entonces no se considera útil y su medida de relevancia será mínima.

Una vez generados los vectores de documentos, al alumno organizará esta información en una matriz de asociación término-documento, utilizando las estructuras de datos necesarias para generar el conjunto completo de índices inversos de los documentos. Esta matriz organiza los documentos por filas y los términos por columnas, de forma que la posición `M[i, j]` de la matriz representa la *relevancia del término de indexación* `j` *en el documento* `i`.

El resultado de esta práctica será un programa que permite eliminar palabras vacías, lematizar y ponderar los términos en función de la frecuencia. Además, generará un índice inverso (estructuras de indización adecuadas) para cada documento, incluyendo la lista completa de términos de indización con la relevancia en el documento.

Recursos:

RECURSOS

- Recursos software.

- Compilador de Borland C++
- Java Development Kit