CAPÍTULO 2

TÉCNICAS ESTADÍSTICAS PARA PROCESAMIENTO DE LENGUAJE NATURAL

La cantidad de texto electrónico disponible en distintos idiomas es enorme. Sin embargo, dichos textos son normalmente "planos" y en muchas ocasiones pueden contener errores tipográficos, ortográficos o gramaticales. A pesar de ello encierran una cantidad enorme de información que los usuarios podrían explotar una vez "tamizada". Para ello es necesario disponer de técnicas de procesamiento de lenguaje natural que permitan la clasificación, categorización, extracción y recuperación de información. Dichas técnicas deberían ser sencillas, robustas y aplicables a múltiples idiomas sin recurrir a conocimiento lingüístico alguno. A lo largo de este capítulo se describirán varias de estas técnicas poniendo énfasis en métodos puramente estadísticos. Así, se describirá el modelo vectorial estudiando su aplicación a la clasificación y categorización de documentos así como a la recuperación de información. Posteriormente se analizará el uso de n-gramas de caracteres en dicho modelo y se continuará con las técnicas Acquaintance y Highlights con las que la propuesta del autor muestra ciertas similitudes.

1 Sobrecarga de información y Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (PLN) es el conjunto de técnicas algorítmicas que tienen como objeto la manipulación y generación de muestras de lenguaje humano tanto en su manifestación escrita como oral. Ejemplos de técnicas de PLN son la generación de habla a partir de texto, el reconocimiento del habla, la traducción automática o la recuperación de información.

En el capítulo anterior se limitó el problema objeto de estudio al procesamiento de texto natural en condiciones "extremas" (grandes volúmenes de texto, múltiples idiomas, ambigüedad, errores tipográficos, ortográficos y gramaticales) por lo que, en este trabajo, no resulta de interés ninguna de las técnicas PLN relacionadas con el habla.

Por otro lado, es posible desglosar el problema de partida, la "sobrecarga de información", en una serie de tareas bien definidas: **categorización** (asignación de un

documento a una categoría previamente conocida), **clasificación** (agrupación de documentos con características similares), **recuperación de información** (localización, dentro de una colección de documentos, de un subconjunto relevante para una consulta formulada por un usuario) y **destilación de información** (extracción de palabras clave, obtención de resúmenes automáticos, respuesta de preguntas, etc.) Así, tampoco se tratarán en este trabajo técnicas PLN que tengan como objeto la creación[1] de muestras de lenguaje.

Aun así existe un amplio repertorio de métodos para resolver cada una de las tareas anteriores y es posible implementar herramientas que utilicen varias de dichas técnicas para afrontar el problema genérico de procesamiento de texto no estructurado en la Web. No obstante, debido a las especiales condiciones en las que dicho procesamiento debería llevarse a cabo el autor considera que las técnicas a emplear deberían verificar los siguientes requisitos:

- Independencia del idioma (aplicables a diversos lenguajes humanos sin cambios o con cambios mínimos).
- Utilización únicamente de métodos estadísticos simples (conocimiento "cero").
- Alta tolerancia al "ruido" (capacidad para trabajar sobre documentos "contaminados[2]" o con errores de cualquier tipo).
- Escalabilidad (posibilidad de ser aplicadas sobre colecciones de documentos muy grandes y de crecimiento continuo).

Por ello no se estudirá ninguna técnica que requiera el uso de "artefactos" lingüísticos como *stemmers*, etiquetado *POS*[3] o desambiguación puesto que éstos requieren conocimientos del idioma en que están escritos los documentos. La razón para esta limitación del abanico de técnicas aceptables es simple: siempre es más sencillo conseguir muestras de texto plano para cualquier lengua que el correspondiente conocimiento lingüístico sobre la misma. Es más, el autor considera indispensable disponer de métodos simples y robustos aplicables en semejantes condiciones como paso previo al desarrollo de técnicas PLN más elaboradas.

A lo largo de este capítulo se analizarán muy brevemente las características de métodos aplicables a cada una de las tareas descritas y que verifican los cuatro primeros requisitos. Sin embargo, a todo lo anterior habría que añadir un quinto requisito, a saber, el uso de una única técnica para afrontar cada una de las tareas anteriores. Recuérdese que el autor sostiene en su tesis no sólo la posibilidad de desarrollar semejante técnica sino que ésta ofrecerá para cada una de las tareas anteriores resultados comparables a los obtenidos

---

[1] Entendiendo la creación como la producción "desde cero" de textos escritos en un lenguaje natural.

[2] Artículos *USENET* a los que no se han eliminado las cabeceras o documentos *HTML* a los que no se ha podido limpiar todo el código *Javascript* son ejemplos de documentos "contaminados".

[3] *Part-Of-Speech* (*POS*) es el papel que una palabra juega en una producción (por ejemplo, sustantivo, verbo, adjetivo, etc.) Un etiquetador *POS* recibe una producción en un lenguaje natural y produce una salida en la que cada palabra está "etiquetada" con uno o más tipos. Por ejemplo, la oración **Él te vino a ver** podría etiquetarse de la siguiente manera: **Él** [Pronombre personal] **te** [Pronombre personal] **vino** [Verbo principal indicativo] **a** [Preposición] **ver** [Verbo principal infinitivo]. En cambio, **El té y el vino están pasados** sería etiquetada de la forma siguiente: **El** [Artículo definido] **té** [Nombre común] **y** [Conjunción coordinada] **el** [Artículo definido] **vino** [Nombre común] **están** [Verbo principal indicativo] **pasados** [Verbo principal participio]. Este tipo de información resulta enormemente útil pero no se puede obtener de manera automática sin conocimiento del lenguaje en cuestión.

con técnicas específicas. Dicha técnica, denominada *blindLight*, se describirá en el siguiente capítulo.

2 El modelo vectorial de documentos

A excepción de la obtención de resúmenes automáticos[1], todas las técnicas PLN que resultan de interés para el problema que nos ocupa y con las que tiene relación la nueva técnica propuesta por el autor tienen dos puntos en común: (1) Requieren una definición de asociación (o similitud) entre dos documentos y (2) precisan una forma de representación de los documentos que permita el cálculo de dicha medida de asociación.

Así, en el caso de la clasificación se desea construir subconjuntos de documentos que exhiban unas características comunes aunque diferentes de las del resto de grupos. Dicho de otro modo, deben encontrarse grupos que maximicen la similitud intragrupal al tiempo que minimicen la similitud intergrupal. La categorización, por su parte, se reduce a determinar qué categoría (que se representará de un modo análogo a los documentos) se encuentra más próxima al documento a categorizar. Por último, los sistemas de recuperación de información reciben consultas (esto es, documentos extremadamente cortos producidos por el usuario) y retornan aquellos documentos de la colección que se encuentran más próximos a las mismas.

Así pues, en lugar de analizar varias técnicas PLN de forma aislada se va a tratar de ofrecer una visión global de las mismas. Para ello se estudiarán las distintas formas en que se puede representar un documento así como las posibles medidas de asociación con cada tipo de representación.

La forma más sencilla de representar un documento es mediante un conjunto de palabras. Aquellas que aparecen en el documento pertenecerán al conjunto y las que no se utilizan, obviamente, no. En este modelo, denominado booleano, las consultas no se representan del mismo modo que los documentos sino bajo la forma de expresiones lógicas que combinan palabras (que presumiblemente se utilizan en los documentos) y los operadores `AND`, `OR` y `NOT`.

```
(information AND retrieval) OR ir
```

Fig. 14 Ejemplo de consulta para un modelo booleano.

El modelo booleano es muy simple, de hecho demasiado: todas las palabras de un documento son consideradas igualmente importantes y las consultas retornan o bien demasiados documentos o bien muy pocos. Por otro lado, puesto que no existe el concepto de similitud no es posible determinar qué documentos satisfacen mejor la consulta, es decir, el funcionamiento es dicotómico: hay documentos que no satisfacen la consulta y otros que sí (aquellos que aparecen en la lista de resultados).

Sin embargo, es posible definir medidas de asociación entre este tipo de representaciones de documentos y, por tanto, similitudes. La más simple de tales medidas es la siguiente:

$$\left|X \cap Y\right| \qquad (1)$$

[1] En realidad, algunas técnicas de extracción de resúmenes se han construido sobre sistemas de recuperación de información y, por tanto, también comparten ambas características.

Que no es más que el número de palabras que aparecen tanto en el documento *X* como en el documento *Y*. Sin embargo, esta medida no tiene en cuenta el número de términos de cada documento y puede resultar engañosa al comparar resultados obtenidos para parejas de documentos con longitudes muy diferentes. Los siguientes coeficientes se basan en el anterior pero incluyen más información sobre los dos documentos a comparar:

$$2\frac{|X \cap Y|}{|X|+|Y|} \quad (2)$$

Coeficiente de Dice

$$\frac{|X \cap Y|}{|X \cup Y|} \quad (3)$$

Coeficiente de Jaccard

$$\frac{|X \cap Y|}{|X| \times |Y|} \quad (4)$$

Coseno

$$\frac{|X \cap Y|}{\min(|X|,|Y|)} \quad (5)$$

Coeficiente de solapamiento

La utilización de tales coeficientes permite mejorar el modelo en varios aspectos. En primer lugar, las consultas pueden representarse como conjuntos de palabras exactamente igual que los documentos. En segundo lugar, es posible obtener un valor numérico que indique cuán fuerte (o débil) es la relación entre dos documentos (y por tanto entre un documento y una consulta).

A pesar de estas ventajas, la representación de los documentos como vectores booleanos no es totalmente satisfactoria puesto que, como ya se dijo, todas las palabras resultan igualmente relevantes lo cual no es realista. Es mucho más conveniente asignar a cada palabra un valor real, un "peso", que indique la importancia de la misma dentro de dicho documento.

Este nuevo modelo, conocido como **modelo vectorial** (Salton y Lesk 1965), considera cada documento de una colección como un vector de pesos en un espacio de *T* dimensiones donde *T* es el número de términos distintos que aparecen en la colección.

$$D_i = (d_{i1}, d_{i2}, d_{i3}, \ldots, d_{it})$$

Fig. 15 Un documento en un espacio vectorial de T dimensiones.

d_{ij} es el peso del término *j-simo* para el documento D_i.

Para calcular el peso de un término en un documento existen distintas alternativas pero en todos los casos se tiene en cuenta lo siguiente:

- La frecuencia de aparición del término en el propio documento, *tf* (Luhn 1957). Los términos que más se repiten en un documento son, en principio, más relevantes que los que se emplean menos.

- El número de documentos de la colección en los que aparece el término, *idf* (Karen Spärck-Jones 1972). Los términos más frecuentes en la colección serán menos relevantes que los más raros.
- La longitud del documento, a fin de garantizar que todos los documentos se comportan de modo similar con independencia de su longitud. En otras palabras, no hay relación entre la relevancia de un documento para una consulta y su longitud.

D1: Microsoft vs Google heats up
D2: Microsoft previews MSN Virtual Earth
D3: MSN to offer virtual Earth map service
D4: MSN joins Google in melding satellite imagery with search
D5: MSN Virtual Earth to take on Google Earth

Q: Google Earth

Fig. 16 Una colección de 5 documentos y una consulta.

Término	IDF	w_{D1i}	w_{D2i}	w_{D3i}	w_{D4i}	w_{D5i}	w_{Qi}
earth	0,33	0	0,33	0,33	0	0,66	0,33
google	0,33	0,33	0	0	0,33	0,33	0,33
heats	1	1	0	0	0	0	0
imagery	1	0	0	0	1	0	0
in	1	0	0	0	1	0	0
joins	1	0	0	0	1	0	0
map	1	0	0	1	0	0	0
melding	1	0	0	0	1	0	0
microsoft	0,50	0,50	0,50	0	0	0	0
msn	0,25	0	0,25	0,25	0,25	0,25	0
offer	1	0	0	1	0	0	0
on	1	0	0	0	0	1	0
previews	1	0	1	0	0	0	0
satellite	1	0	0	0	1	0	0
search	1	0	0	0	1	0	0
service	1	0	0	1	0	0	0
take	1	0	0	0	0	0,50	0
to	0,50	0	0	0,50	0	0,50	0
up	1	1	0	0	0	0	0
virtual	0,33	0	0,33	0,33	0	0,33	0
vs	1	1	0	0	0	0	0
with	1	0	0	0	1	0	0

Fig. 17 La colección anterior y la consulta representadas en un espacio vectorial.

El valor *idf* se ha simplificado como el inverso del número de documentos en que aparece cada término. El peso de cada término para cada documento es el producto de dicho *idf* por la frecuencia de aparición del término en el documento, así, el término `Earth` tiene un peso de 0.33 en todos los documentos a excepción del quinto pues en éste aparece dos veces y, en consecuencia, el peso es 0.66.

No parece necesario entrar en mayores detalles acerca del cálculo de los pesos para entender el funcionamiento del modelo: para cada documento de una colección se genera un vector de valores reales y en el caso de los sistemas de recuperación de información basados en el modelo vectorial se procede del mismo modo para las consultas recibidas por el sistema.

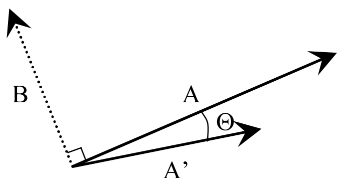
Así pues, dados dos vectores de pesos es posible, de manera análoga a como se hacía con vectores booleanos, obtener una medida numérica de su asociación. Para ello, pueden adaptarse algunas de la medidas de asociación mostradas antes, siendo una de las más populares la denominada **"función del coseno"** (véase Fig. 18).

$$\frac{\sum_{i=1}^{n} q_i \cdot d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \cdot \sqrt{\sum_{i=1}^{n} d_i^2}}$$

Fig. 18 Ecuación de la función del coseno entre los documentos *Q* y *D*.

q_i y d_i son el componente i-ésismo de los vectores *Q* y *D*, respectivamente. *n* es el número de términos distintos en la colección.

Esta medida admite una interpretación geométrica (véase Fig. 19) puesto que su valor numérico puede considerarse como el coseno del ángulo formado por los vectores de los documentos comparados. Así, un valor 0 implica que los vectores son ortogonales (esto es, no son similares) mientras que un valor 1 significa que los vectores forman un ángulo de 0º (es decir, son iguales o, más bien, muy parecidos).



d (A, A') ≅ 1 ➔ Θ ≅ 0º
d (A, B) = 0 ➔ Θ = 90º

Fig. 19 Interpretación geométrica de la función del coseno en un espacio bidimensional.

Los documentos A y A' tienen una similitud próxima a 1, es decir, forman un ángulo cercano a 0º y, por tanto, "apuntan" en la misma dirección. En cambio, la similitud entre A y B es 0, es decir, son ortogonales.

Término	w_{D1i}^2	w_{D2i}^2	w_{D3i}^2	w_{D4i}^2	w_{D5i}^2	w_{Qi}^2	$w_{D1i} \cdot w_{Qi}$	$w_{D2i} \cdot w_{Qi}$	$w_{D3i} \cdot w_{Qi}$	$w_{D4i} \cdot w_{Qi}$	$w_{D5i} \cdot w_{Qi}$
earth	0	0,11	0,11	0	0,44	0,11	0	0,11	0,11	0	0,22
google	0,11	0	0	0,11	0,11	0,11	0,11	0	0	0,11	0,11
heats	1	0	0	0	0	0	0	0	0	0	0
imagery	0	0	0	1	0	0	0	0	0	0	0
in	0	0	0	1	0	0	0	0	0	0	0
joins	0	0	0	1	0	0	0	0	0	0	0
map	0	0	1	0	0	0	0	0	0	0	0
melding	0	0	0	1	0	0	0	0	0	0	0
microsoft	0,25	0,25	0	0	0	0	0	0	0	0	0
msn	0	0,06	0,06	0,06	0,06	0	0	0	0	0	0
offer	0	0	1	0	0	0	0	0	0	0	0
on	0	0	0	0	1	0	0	0	0	0	0
previews	0	1	0	0	0	0	0	0	0	0	0
satellite	0	0	0	1	0	0	0	0	0	0	0
search	0	0	0	1	0	0	0	0	0	0	0
service	0	0	1	0	0	0	0	0	0	0	0
take	0	0	0	0	0,25	0	0	0	0	0	0
to	0	0	0,25	0	0,25	0	0	0	0	0	0
up	1	0	0	0	0	0	0	0	0	0	0
virtual	0	0,11	0,11	0	0,11	0	0	0	0	0	0
vs	1	0	0	0	0	0	0	0	0	0	0
with	0	0	0	1	0	0	0	0	0	0	0

|D1|	|D2|	|D3|	|D4|	|D5|	|Q|
1,83	1,24	1,88	2,68	1,49	0,47

D1·Q	D2·Q	D3·Q	D4·Q	D5·Q
0,11	0,11	0,11	0,11	0,33

|D1|·|Q|	|D2|·|Q|	|D3|·|Q|	|D4|·|Q|	|D5|·|Q|
0,86	0,58	0,88	1,26	0,70

D1, Q	D2, Q	D3, Q	D4, Q	D5, Q
0,13	0,19	0,13	0,09	0,47

Fig. 20 Cálculo de la función del coseno entre la consulta y los documentos anteriores.

? Google Earth

(0,47) **D5:** MSN Virtual Earth to take on Google Earth
(0,19) **D2:** Microsoft previews MSN Virtual Earth
(0,13) **D1:** Microsoft vs Google heats up
(0,13) **D3:** MSN to offer virtual Earth map service
(0,09) **D4:** MSN joins Google in melding satellite imagery with search

Fig. 21 Lista ordenada de resultados al realizar la consulta sobre la colección.

Otra ventaja de la función del coseno radica en el hecho de que no es necesario normalizar los pesos de los términos en función de la longitud de los documentos puesto que lo que se "mide" es el ángulo formado por los vectores (véase Fig. 19).

Mediante la utilización de vectores de pesos para representar documentos (y consultas) y el uso de la función del coseno u otra similar es sencillo implementar sistemas de recuperación de información (véanse Fig. 16, Fig. 17, Fig. 20 y Fig. 21). Representando de manera vectorial las características comunes de un conjunto de documentos se puede utilizar este modelo también para realizar categorización y clasificación. En el modelo booleano resulta muy sencillo encontrar un nuevo vector con las características comunes de dos o más documentos, basta con realizar la intersección de todos ellos para obtener un nuevo vector que contenga aquellos términos comunes a todos ellos. En el caso del modelo vectorial la solución es similar: se necesita calcular el **centroide** del conjunto de vectores, es decir, obtener un nuevo vector de pesos donde cada peso será la media de los pesos de los distintos vectores del conjunto.

En definitiva el modelo vectorial ofrece un modo de representar documentos y consultas mediante vectores de pesos, una serie de medidas para determinar la asociación entre dichos vectores y un modo de calcular un nuevo vector (centroide) para representar características comunes de un conjunto de vectores (o lo que es lo mismo, un grupo de documentos). Todo ello permite aplicar el modelo vectorial a tres de las cuatro tareas antes mencionadas[1]:

- **Categorización:** un sistema basado en el modelo vectorial puede ser "entrenado" de una forma muy sencilla. Dado un conjunto de documentos de entrenamiento etiquetados se calcula el centroide del conjunto de documentos pertenecientes a cada categoría. Una vez hecho esto, la categorización resulta tan sencilla como determinar qué centroide (categoría) es el más próximo al vector del documento a categorizar.
- **Clasificación:** dado el conjunto de vectores para los documentos a clasificar y una medida de asociación es posible implementar cualquiera de los algoritmos clásicos de clasificación (*clustering* aglomerativo, partición, *k*-vecinos, etc.)
- **Recuperación de información:** se obtiene un conjunto de vectores para todos los documentos de la colección, al recibir una consulta se genera un vector para la consulta y se calcula la similitud entre éste y los vectores de la colección proporcionando como resultado una lista de documentos ordenada inversamente por su similitud a la consulta.

[1] Y también a la extracción de resúmenes: (1) cada sentencia de un documento se representa mediante un vector, (2) se calcula el centroide de estos vectores, (3) se calcula la distancia coseno de cada vector sentencia al centroide y (4) se extraen las sentencias más próximas al centroide como resumen del texto. No obstante, esta técnica, aunque sencilla, no es la mejor posible para obtener resúmenes extractivos.

Además, estas implementaciones cumplen tres de los cuatro requisitos deseables para resolver el problema del tratamiento de texto no estructurado en la Web:

- **Independencia del idioma:** el modelo vectorial es aplicable a toda clase de idiomas siempre que puedan extraerse palabras de los documentos (sencillo en muchos idiomas pero más complejo en otros, como el chino o el japonés, en los que no se separan las palabras).
- **Utilización únicamente de métodos estadísticos simples:** el modelo vectorial cumple totalmente este requisito al calcularse los pesos de los términos a partir de datos extraíbles directamente de los textos sin ningún tipo de conocimiento lingüístico[1].
- **Escalabilidad:** el modelo vectorial cumple este requisito parcialmente puesto que al necesitarse una fase de "indexado" en la que se generen los vectores de todos los documentos de la colección no es posible disponer de colecciones que crezcan de manera continua sino que, en ciertas ocasiones, es necesario "detener" el sistema y volver a generar toda la información del mismo (recuérdese que para calcular el peso de un término es necesario conocer en cuántos documentos se utiliza dicho término).

Por otro lado, el requisito de alta tolerancia al ruido es difícilmente alcanzable con implementaciones del modelo vectorial que utilizan palabras como términos. Puesto que palabras distintas constituyen términos distintos y, por tanto, coordenadas diferentes, los errores tipográficos, la utilización inconsistente de marcas diacríticas o simples faltas de ortografía influirán en los resultados obtenidos. Esto puede solucionarse parcialmente con algoritmos de *stemming* pero, de nuevo, se trata de conocimiento lingüístico.

3 Utilización de *n*-gramas en el modelo vectorial

En el apartado anterior se describió el modelo vectorial de manera sencilla mostrando su aplicabilidad a tareas de categorización, clasificación y recuperación de información. Se afirmó que dicho modelo, aunque aplicable a multitud de idiomas, no es excesivamente tolerante al ruido si se emplean palabras como términos. Sin embargo, el modelo no especifica qué elementos de un documento deben utilizarse como términos, es decir, no exige que se empleen palabras y admite otras posibilidades.

Una de las modificaciones más sencillas fue utilizada por Salton (1968) y consiste en utilizar no palabras sino versiones "reducidas" de las mismas tras aplicar un algoritmo de *stemming*. Esta modificación permite, en cierta medida, reducir el número de términos y "fusionar" algunos relacionados semánticamente entre sí. Sin embargo, y dejando a un lado el hecho de que hay que construir un *stemmer* para cada idioma, esta técnica no mejora el comportamiento frente al ruido (véase Fig. 22).

Una alternativa mucho más sencilla, puesto que no requiere implementar algoritmos específicos para cada idioma, y que es mucho más tolerante a errores[2] en el texto consiste en

[1] En realidad, la mayoría de implementaciones utilizan las ya mencionadas listas de "palabras vacías" que suponen el uso de conocimiento lingüístico. No es este el caso de la técnica propuesta por el autor.

[2] La utilidad de los *n*-gramas para enfrentarse a texto "ofuscado" quedó patente durante la *Confusion Track* del *TREC-5* (*Text REtrieval Conference*, Congreso sobre Recuperación de Texto). El objetivo de esa tarea era recuperar documentos para los que se disponía de versiones con ruido (5 y 20%) debido a un escaneado de baja resolución. La mayoría de participantes emplearon *n*-gramas de un modo u otro para afrontar la tarea obteniendo resultados muy satisfactorios (Kantor y Voorhees 2000).

el uso de *n*-gramas. De manera genérica, un ***n*-grama** es una secuencia de *n* elementos, palabras o caracteres, extraídos de un texto de forma no necesariamente correlativa. Sin embargo, se entiende habitualmente que un *n*-grama es una secuencia de *n* caracteres contiguos que puede contener blancos[1] y, por tanto, estar formado por segmentos de varias palabras consecutivas.

```
* busineses → busines
  businesses → busi
     busy → busi
 * desgined → desgin
  designed → design
   design → design
```

Fig. 22 Palabras inglesas procesadas con el algoritmo de *stemming* de Porter.

Se muestran en negrita aquellas palabras a las que el *stemmer* asocia la misma forma reducida, precedidas de un asterisco se presentan versiones con errores tipográficos. Obsérvese que la forma reducida asignada por el algoritmo no es la misma.

El uso de *n*-gramas en tareas de recuperación de información tiene una tradición de, al menos, 30 años. Durante este tiempo se han implementado múltiples modelos con diferencias de planteamiento muchas veces sutiles. A fin de esbozar someramente la trayectoria que se ha seguido en este campo de investigación se hará referencia a los trabajos realizados por Barton *et al.* (1974), D'Amore y Mah (1985), Cavnar (1994) y McNamee y Mayfield (2004).

```
* busineses → _bus, busi, usin, sine, ines, nese, eses, ses_
 businesses → _bus, busi, usin, sine, ines, esse, sses, ses_
                busy → _bus, busy, usy_
   * desgined → _des, desg, esgi, sgin, gine, ined, ned_
      designed → _des, desi, esig, sign, igne, ned_
         design → _des, desi, esig, sign, ign_
```

Fig. 23 4-gramas obtenidos para una serie de palabras inglesas.

Se muestran precedidas de un asterisco las versiones con errores tipográficos y en negrita aquellos 4-gramas comunes a varias de las palabras. Obsérvese cómo aquellas palabras que serían "fusionadas" por un *stemmer* comparten un gran número de *n*-gramas y que las formas incorrectas comparten varios *n*-gramas con las formas correctas. Se representa el blanco por un guión bajo.

Barton *et al.* (1974) analizaron la forma de obtener, de modo automatico, *n*-gramas de longitud variable de tal modo que su frecuencia de aparición en un ***corpus*** adecuado fuese similar: mayor o igual que un umbral fijado empíricamente (véase Fig. 24). Esto tenía como principal objetivo reducir el número de "índices" a utilizar en un diccionario de términos al tiempo que se garantizaba que dichos términos eran los más frecuentemente utilizados en los documentos[2].

La colección indexada de documentos se representaba como una matriz de bits donde cada columna representaba un documento y cada fila un término. Los bits activos indicaban la aparición del término en el correspondiente documento. Para realizar una consulta simplemente se debía obtener un vector de bits para dicha consulta y determinar qué documentos presentaban más bits en común (véase Fig. 25).

En este sentido, podría considerarse que Barton *et al.* implementaron un modelo booleano basado en *n*-gramas de longitud variable. Por otro lado, el interés fundamental de su investigación radicaba en aspectos tales como la eficiencia espacial y temporal por lo que,

[1] Algunos investigadores no sólo incluyen blancos sino cualquier tipo de símbolo de puntuación.

[2] En realidad en los títulos de los documentos, sin embargo, conceptualmente este detalle es irrelevante.

aunque constatan unos resultados satisfactorios, no hacen ninguna comparación con otros sistemas.

Corpus: Using direct access computer files of bibliographic information, an attempt is made to overcome one of the problems often associated with information retrieval, namely, the maintenance and use of large dictionaries, the greater part of which is used only infrequently.

Índices usando *n*-gramas de tamaño máximo 5 y con una frecuencia absoluta igual o superior a 3:

the	_inf	_of_	the_	tion	e_o	he_	inf	ion	n_a	of_
_a	_o	an	ar	at	d_	e_	en	er	es	f_
ic	ma	na	nf	on	re	s_	t_	te	us	_
a	b	c	d	e	f	g	h	i	l	m
n	o	p	q	r	s	t	u	v	w	y

Índices usando *n*-gramas de tamaño máximo 5 y con una frecuencia absoluta igual o superior a 4:

_of_	of_	_a	_i	_o	e_	f_	in	io	ma	n_
on	re	s_	te	th	_	a	b	c	d	e
f	g	h	i	l	m	n	o	p	q	r
s	t	u	v	w	y					

Fig. 24 Índices extraídos para un *corpus* mínimo según el método de Barton *et al.* (1974).

El objetivo básico de la técnica de Barton *et al.* (1974) era reducir el número de índices necesarios para representar documentos y consultas. Su método requiere dos parámetros: el máximo tamaño deseado para los n-gramas y la frecuencia umbral que deben superar en el *corpus* los *n*-gramas para considerarse índices. Según los propios autores este último parámetro debe determinarse de manera empírica. Nótese que el número de índices es inversamente proporcional a la frecuencia umbral seleccionada y que se incluyen entre los índices todos los caracteres individuales que aparecen en el *corpus*. En aras de la claridad se ha sustituido el espacio en blanco por el guión bajo.

Raymond D'Amore y Clinton P. Mah (1985) desarrollan un método situado a medio camino entre la propuesta de Barton *et al.* (1974) y una implementación del modelo vectorial basada en *n*-gramas de caracteres. Al igual que Barton *et al.*, su principal objetivo es reducir el número de índices para lo cual emplean *n*-gramas de caracteres aproximadamente equipobrables en un *corpus* de referencia (no necesariamente coincidente con la colección de documentos a indexar).

Consulta: An information-theoretic approach to text searching in direct access systems

Representación de la consulta usando los índices "5/3" (con información redundante):

_inf	tion	inf	ion	_a	an	ar	at	es	ic	ma
nf	on	re	s_	t_	te	-	_	a	c	d
e	f	g	h	i	m	n	o	p	r	s
t	x	y								

Representación de la consulta usando los índices "5/3" (sin información redundante):

_inf	tion	_a	an	ar	at	es	ic	ma	re	s_
t_	te	-	_	a	c	d	e	g	h	i
m	n	o	p	r	s	t	x	y		

Representación de la consulta usando los índices "5/4" (con información redundante):

_a	_i	in	io	ma	n_	on	re	s_	te	th
-	_	a	c	d	e	f	g	h	i	m
n	o	p	r	s	t	x	y			

Representación de la consulta usando los índices "5/4" (sin información redundante):

_a	_i	f_	in	io	ma	re	s_	te	th	-
_	a	c	d	e	f	g	h	i	m	n
o	p	r	s	t	x	y				

Fig. 25 Consulta representada mediante índices extraídos según el método de Barton *et al.* (1974).

Las consultas se representan empleando los índices extraídos previamente del *corpus*. La consulta puede contener información redundante (es decir, *n*-gramas que se solapan) o no. Una vez obtenida la consulta la comparación con los documentos es booleana.

D'Amore y Mah proponen utilizar conjuntos con un número fijo de índices que serían *n*-gramas de distintas longitudes aunque, fundamentalmente, emplean 2- y 3-gramas. A diferencia de Barton *et al.* no proporcionan un método automático para la obtención de tales *n*-gramas puesto que, afirman, deben determinarse experimentalmente para cada aplicación. Según estos investigadores un conjunto de índices completo (para textos en inglés) contendría aproximadamente 6.500 *n*-gramas: todos los 2-gramas alfanuméricos (36x36) junto con 200x26 3-gramas alfabéticos. Dichos 3-gramas se obtendrían "extendiendo" los 200 2-gramas alfabéticos más frecuentes en inglés y es precisamente esa selección de 3-gramas la que requiere un análisis de la colección[1]. Una vez seleccionados los índices se determina para cada *n*-grama *i* su peso w_i:

$$w_i = \frac{1}{\sqrt{p_i}} \equiv \frac{1}{\sqrt{N_i / N}} = \sqrt{\frac{N}{Ni}}$$

Donde p_i es la probabilidad de aparición del *n*-grama *i* en el *corpus* de referencia que D'Amore y Mah calculan como el cociente entre el número de documentos que contienen el *n*-grama *i*, N_i, y el número total de documentos en el *corpus*, *N*. De este modo, el peso que dichos autores asignan a cada *n*-grama del conjunto de índices es, conceptualmente, muy similar a la aplicación de *idf* (Spärck-Jones 1972) –véase ecuación en página 137.

Por su parte, cada documento es representado mediante un vector de *n*-gramas tomados del conjunto de índices a los que se asigna su frecuencia relativa de aparición en el documento. Para calcular la similitud entre dos documentos (o entre un documento y una consulta) *d* y *q* se debe calcular su producto escalar aunque introduciendo en dicho cálculo el peso fijado *a priori* para cada *n*-grama:

$$S(d,q) = \sum_x w_x \cdot f_x^d \cdot f_x^q$$

Donde w_x es el peso del *n*-grama *x* en el *corpus* de referencia y f_x^d y f_x^q son las frecuencias relativas de aparición de *x* en los documentos *d* y *q*, respectivamente. Nótese que este modo de calcular la similitud interdocumental es semejante al uso de un esquema de ponderación *tf*idf*.

Así, esta propuesta es relativamente similar a una implementación del modelo vectorial basada en *n*-gramas con las salvedades de no emplear todos los *n*-gramas posibles sino un número reducido, utilizar como medida de asociación el producto escalar en lugar de la función del coseno y aplicar un esquema de ponderación ligeramente distinto del *tf*idf* "tradicional".

Existe, no obstante, otro aspecto en que la técnica de D'Amore y Mah difiere del modelo vectorial y es el uso de un valor umbral para diferenciar las similitudes significativas de las no significativas (casuales). Para ello es necesario determinar la similitud mínima esperable en el *corpus* de referencia para tomar en consideración sólo aquellos valores de similitud que superan este valor mínimo. Dicho umbral se calcularía de este modo:

$$\sum_x w_x \cdot p_x \cdot p_x = \sum_x w_x \cdot p_x^2 = \sum_x \frac{1}{\sqrt{p_x}} \cdot p_x^2 = \sum_x p_x^{3/2}$$

[1] Por ejemplo, dado el 2-grama `th`, ¿qué 3-gramas se escogerían como índices? ¿`the`, `thy`, `ith`, …?)

Donde w_x y p_x son, respectivamente, el peso y la probabilidad del *n*-grama *x* en el *corpus* de referencia.

En resumen, D'Amore y Mah desarrollan un sistema próximo al modelo vectorial pero que emplea un conjunto fijo de 2- y 3-gramas de caracteres como términos, el producto escalar como medida de asociación, una modificación del esquema de ponderación *tf*idf* tradicional y sólo considera como significativos (no casuales) aquellos valores de similitud que superan un valor mínimo esperable. D'Amore y Mah señalan que su técnica ofrece resultados próximos a los sistemas de recuperación de información en texto completo aunque no proporcionan ninguna evaluación de dicho rendimiento.

Posteriormente, William B. Cavnar (1994) implementó un sistema fiel al modelo vectorial empleando *n*-gramas como términos en lugar de palabras o *stems*. Cavnar obtuvo resultados análogos a los de D'Amore y Mah que venían a confirmar que los *n*-gramas podían competir, en cuanto a resultados, con otros sistemas que utilizaban palabras como términos y que, a diferencia de su sistema, requerían el uso de "artefactos" como *stemming* o eliminación de palabras vacías.

Más recientemente, el sistema *HAIRCUT*[1] desarrollado en la universidad Johns Hopkins ha probado nuevamente que la utilización (sola o combinada) de técnicas "ligeras" (sin utilización de conocimientos lingüísticos previos), entre las que se incluye el uso de *n*-gramas, y métodos estadísticos pueden resultar *"al menos tan efectivos como enfoques que utilizan tratamientos dependientes del idioma y quizás más"* (McNamee y Mayfield 2004).

Así pues, puede afirmarse que la utilización de *n*-gramas para tareas de recuperación de información con independencia del modelo teórico[2] y del idioma sobre los que se apliquen proporciona resultados comparables a los obtenidos con técnicas adaptadas a cada idioma. *blindLight* enlaza con esta línea de investigación aunque, como se verá más adelante, se diferencia en algunos aspectos importantes.

Por otro lado, las aplicaciones de los *n*-gramas van más allá de la recuperación de información y, de hecho, se han aplicado a todas las tareas mencionadas al comienzo del capítulo, específicamente a la categorización y clasificación de documentos, así como a la extracción automática de términos clave.

En este sentido requieren mención especial los trabajos realizados por Marc Damashek (1995) y Jonathan D. Cohen (1995) el primero con la técnica *Acquaintance* y el segundo con *Highlights*. Puesto que la nueva técnica propuesta por el autor de este trabajo establece, en cierta medida, un puente entre las desarrolladas por Cohen y Damashek es necesario describir éstas brevemente antes de poder presentar los fundamentos de *blindLight* y señalar las diferencias entre la propuesta del autor y el resto de métodos desarrollados hasta la fecha.

3.1 Estimación de la similitud interdocumental utilizando *n*-gramas (*Acquaintance*)

La propuesta de Damashek (1995) presenta semejanzas con la de D'Amore y Mah (1985). Al igual que ellos, *Acquaintance* representa los documentos como vectores de pesos de *n*-gramas donde cada peso es la frecuencia relativa de aparición del correspondiente

[1] `http://haircut.jhuapl.edu/index.html`

[2] Es posible desarrollar sistemas de recuperación de información que usen *n*-gramas como términos y que implementen el modelo vectorial, el probabilístico o cualquier otro tipo de modelo *ad hoc*. En este sentido la aplicación actual de *blindLight* a esta tarea, aunque similar en ciertos aspectos al modelo vectorial, debería considerarse como un modelo diferente.

n-grama en el documento y también utiliza el producto escalar como métrica para comparar los vectores.

No obstante, Damashek no utiliza *n*-gramas de tamaño variable ni tampoco establece un conjunto de *n*-gramas de tamaño fijo para realizar el indexado. Además, Damashek no sólo emplea la técnica *Acquaintance* para llevar a cabo recuperación de información sino que la extiende para realizar categorización y clasificación de documentos. Este paso resulta natural puesto que, como se mostró en el apartado dedicado al modelo vectorial, estas tareas pueden llevarse a cabo fácilmente si se dispone de una métrica de la similitud entre documentos y de un modo de obtener centroides para conjuntos de documentos, ambas cosas posibles con *Acquaintance*.

Hay que señalar, sin embargo, dos debilidades en esta técnica. En primer lugar, el rendimiento obtenido al realizar recuperación de información empleando consultas cortas (las más habituales en la Web) es, en palabras del propio Damashek, pobre. *Acquaintance* ofrece, en cambio, mejores resultados cuando se recupera información partiendo de un documento de ejemplo. En segundo lugar, Damashek (1995) afirma también lo siguiente:

> *La métrica [de similitud entre documentos] falla en tareas más sutiles como la discriminación basándose en el asunto [del documento] puesto que los vectores obtenidos a partir de texto plano están habitualmente dominados por componentes no-informativas (por ejemplo, en inglés los n-gramas derivados de "is the", "and the", "with the", …) y son estos componentes de mayor peso los que más influyen en el producto escalar.*

Para enfrentarse a este problema Damashek sugiere realizar una traslación del conjunto de documentos sobre los que van a realizarse las medidas de similitud a fin de minimizar la influencia de estos componentes "no-informativos". Para ello, propone sustraer, componente a componente, el centroide de cada elemento del conjunto. Esto puede realizarse para toda la colección de documentos (lo que equivaldría en cierta medida a eliminar palabras vacías) o para subconjuntos de la misma obtenidos tras una clasificación.

En resumen, *Acquaintance* es una implementación del modelo vectorial en la que se obtiene un vector de pesos de *n*-gramas para cada documento de la colección. Dichos pesos no son más que la frecuencia relativa de aparición de cada *n*-grama en el documento en cuestión. Para evaluar la similitud entre dos vectores se calcula el producto escalar de los mismos y a fin de evitar la influencia de componentes de peso elevado pero poca o nula capacidad discriminativa se traslada el conjunto de documentos sustrayendo de cada vector el centroide de la colección. De este modo, es posible llevar a cabo recuperación de información empleando documentos de ejemplo así como clasificación y categorización.

3.2 Extracción automática de términos clave utilizando *n*-gramas (*Highlights*)

El objetivo fundamental de *Highlights* (Cohen 1995) es extraer de manera automática unos pocos términos clave[1] que permitan a un usuario determinar con rapidez el asunto tratado en un documento. Esta técnica permitiría, por ejemplo, decidir qué documentos obtenidos como respuesta a una consulta son verdaderamente relevantes y cuáles no. Cohen, al igual que el autor de este trabajo, estableció una serie de requisitos para su técnica: debía ser independiente del lenguaje, del dominio de conocimiento y estar basada tan sólo en métodos estadísticos. Para ello, utilizó *n*-gramas de caracteres y vectores de *n*-gramas para representar tanto los documentos como la colección.

[1] Es necesario notar que los términos clave podían ser no sólo palabras sino frases (por ejemplo, `recuperación de información`, `interfaz de usuario`, `lenguajes de programación`, etc.)

Highlights utiliza vectores de pesos de *n*-gramas de manera similar a como se utilizaban en *Acquaintance*. No obstante, como fase previa al cálculo de los vectores de los distintos documentos era necesario construir un "contexto" dentro del cual se realizaría el posterior procesamiento. Dicho contexto no era más que un vector obtenido al tratar toda la colección como un único texto de gran longitud[1]. Los vectores de los distintos documentos eran comparados entonces con este contexto para determinar qué *n*-gramas de cada documento eran menos "probables" con relación a la colección o lo que es lo mismo, menos comunes y, por tanto, más interesantes para describir el documento en cuestión.

Finalmente, una vez establecidos los pesos para cada *n*-grama de un documento *Highlights* procedía a "puntuar" los caracteres del texto. Para ello cada vez que un *n*-grama aparecía en el documento se asignaba su puntuación al carácter central del mismo (véase Fig. 26). Al finalizar esta fase se establecía un umbral que separaba los caracteres "interesantes" de los "no interesantes". Mediante este umbral se extraían aquellas palabras o frases que incluyesen algún carácter relevante. Estos términos eran incluidos en la lista final de resultados que sólo necesitaba ser ordenada antes de ser ofrecida al usuario.

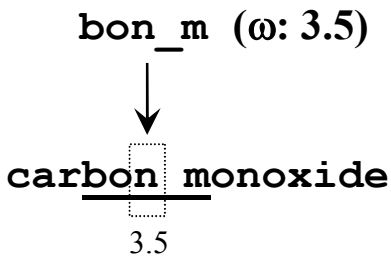


Fig. 26 El peso de un *n*-grama es asignado al carácter central del mismo.

Highlights "puntua" los caracteres que aparecen en un documento a fin de localizar los términos clave. Para ello se localizan las apariciones de cada *n*-grama en el texto y se asigna su peso al carácter central del *n*-grama.

4 Obtención de resúmenes automáticos

Al comienzo del capítulo se señaló que el problema de procesar texto en la Web podía dividirse en un conjunto de tareas tales como: categorización, clasificación, recuperación de información y extracción de información. En los apartados anteriores se han analizado las características fundamentales de las tres primeras tareas y se ha visto que, debido a su estrecha relación, todas pueden ser resueltas empleando prácticamente las mismas técnicas.

Por otro lado, teniendo en cuenta las características del texto disponible libremente en la Web se puso énfasis en aquellas técnicas que mostrasen una alta tolerancia al ruido y empleasen métodos estadísticos simples. Así, se vio cómo la utilización de *n*-gramas[2] resulta particularmente interesante.

Se mostraron seguidamente dos técnicas especialmente relevantes basadas en el uso de tales *n*-gramas. La primera, *Acquaintance* (Damashek 1995), facilitaba la comparación de documentos permitiendo la categorización, clasificación y recuperación de información. La

[1] La idea guarda similitudes con el uso de centroides en *Acquaintance* pero su construcción es totalmente diferente.

[2] Entendidos éstos como secuencias de caracteres, espacios en blanco incluidos, extraídos del texto de manera correlativa.

segunda, *Highlights* (Cohen 1995), tenía como objetivo la extracción de una serie de términos clave (palabras e incluso frases) que permitiesen que un usuario determinase con facilidad el asunto tratado en un documento diferenciándolo del resto de documentos de su entorno.

Esta última es la más próxima a la cuarta tarea de interés para aliviar la sobrecarga de información en la Web: el resumen automático. Sin embargo, no puede considerarse en modo alguno que la resuelva ya que *Highlights* se limita a proporcionar una lista ordenada de términos relevantes para un documento y nunca una versión resumida del mismo.

Puesto que uno de los objetivos de la nueva técnica presentada en este trabajo es la obtención automática de un resumen a partir de un único documento[1], en este apartado se describirán a grandes rasgos los aspectos más relevantes del campo. Se ofrecerán más detalles en el capítulo 7.

Luhn, al que ya se citó como uno de los pioneros en el campo del tratamiento automático de textos, fue el primero en proponer un sistema para obtener un resumen de un documento empleando medios mecánicos (Luhn 1958). Los resúmenes construidos por su sistema eran extractivos, genéricos e informativos[2]. Extractivos puesto que el resumen consistía en una selección del texto original. Genéricos al construírse siempre del mismo modo, reflejando el punto de vista del autor del documento sin permitir que el usuario los orientase para satisfacer posibles consultas. E informativos puesto que incluían los contenidos más relevantes del original en lugar de describir la naturaleza del texto.

No obstante, es posible obtener resúmenes por abstracción, esto es, el sistema no extrae sentencias del texto original sino que "redacta" un documento completamente nuevo. Asimismo, los resúmenes pueden adaptarse a los requisitos que el usuario especifique en forma de consulta. O pueden ser descriptivos, es decir, sin reflejar los contenidos del documento original pueden indicar la naturaleza del mismo.

Es necesario decir que el resumen automático por abstracción así como la construcción de resúmenes descriptivos requieren de técnicas mucho más sofisticadas que para la "simple" extracción de información relevante; de hecho *"no es probable que se construyan sistemas prácticos de resumen por abstracción en el futuro cercano"* (Hovy 1999, p. 7). Por otro lado, generar resúmenes adaptados al usuario siempre puede afrontarse como una tarea de recuperación de información en la que cada sentencia del documento es tratada como un documento individual.

A la hora de afrontar la tarea de obtener resúmenes automáticos el autor de este trabajo se ha centrado únicamente en la construcción de resúmenes extractivos, genéricos e informativos. Por otro lado, como ya se dijo con anterioridad y se verá más adelante, la nueva técnica propuesta, *blindLight*, utiliza *n*-gramas de caracteres no sólo para las tareas de categorización, clasificación y recuperación de información sino también para la construcción de los resúmenes. Es por ello que se afirmaba que establecía un vínculo entre *Acquaintance* (Damashek 1995) y *Highlights* (Cohen 1995).

Hasta donde sabe el autor, tan sólo ha habido hasta la fecha otro intento de obtener resúmenes extractivos empleando *n*-gramas de caracteres. Joel Larocca Neto *et al.* (2000) construyeron un sistema para generar resúmenes extractivos adaptando la técnica de ponderación *tf*idf* a colecciones de sentencias en lugar de colecciones de documentos. Así,

---

[1] Es decir, el único *"corpus"* utilizado para la obtención de información estadística acerca del lenguaje utilizado es el propio documento a resumir.

[2] Las definiciones para resúmenes por extracción/abstracción, informativo/descriptivo y genérico/adaptado al usuario son las recogidas por Eduard Hovy (1999).

proponían una nueva medida, *tf*isf*, para ponderar los términos de un texto donde *isf* es el número de sentencias que incluyen un término dado. El resumen se construye con aquellas sentencias con un valor *tf*isf* más elevado. Los *n*-gramas de caracteres son uno de los dos tipos de términos que pueden utilizarse en su sistema, siendo el segundo palabras individuales una vez eliminadas aquellas vacías de contenido y aplicado un *stemmer*.

Así pues, la novedad de la propuesta de Neto *et al.* es relativa puesto que la utilización de *n*-gramas como términos de modelos vectoriales es bien conocida y ya se ha descrito en apartados anteriores. Por otro lado, cuando se describa a lo largo del resto de la disertación la técnica *blindLight* se podrá comprobar que la semejanza entre ambas propuestas se limita a la utilización de *n*-gramas de caracteres puesto que la forma en que se calculan los pesos de dichos *n*-gramas, se utilizan para representar los documentos, se calculan las similitudes, así como la manera en que se construyen los resúmenes son totalmente diferentes.