Lematización para palabras médicas complejas: Implementación de un algoritmo en LISP.

PEINADO RODRIGUEZ Jesús*

SUMMARY

The automated retrieval of files depends critically on the ability to generate precise signs concepts. Stemming is a technique very useful but medical terms are very complex terms which need special attention. Our main was to develop a modular algorithm for complex medical terms in order to follow a new space of research on Information retrieval. The algorithm was developed using LISP, a programming language, and exhaustive controlled list of rules. As a result, we found good precision with less recall when we asked for concepts saving signs concepts for each medical term. *(Rev Hed Hered 2003; 14:224-229).*

Key words: LISP, Stemming, medical terms, information retrieval.

INTRODUCCIÓN

Lematización (Stemming en Ingles) es una técnica en la recuperación de datos en los sistemas de información (RDSI), esta técnica sirve para reducir variantes morfológicas de la formas de una palabra a raíces comunes o lexemas; con el fin de mejorar la habilidad de los motores de búsqueda para mejorar las consultas en documentos. Básicamente, este consiste en remover el plural, el tiempo, o los atributos finales de la palabra (1).

La aplicación estándar de esta técnica en RDSI, esta en la recuperación precisa de documentos sin redundar en nuevas búsquedas (2).

Por otro lado, desde el inicio de la estandarización y uso de sistemas de información se viene trabajando en la perfección de estos sistemas, uno de los principales objetivos de estos procesos es la ambigüedad en los conceptos, para lo cual surgió el lenguaje formal o el uso de números en el lenguaje. El lenguaje formal es muy importante cuando necesitamos eliminar la ambiguedad de cualquier vocabulario, en este sentido un uso es a manera de índices. Existen 25 vocabularios médicos con diverso alcance del lenguaje formal, uno de ellos es el código internacional de enfermedades versión 10 CIE-10 (3), en el cual los conceptos médicos referidos al diagnostico tienen un correspondiente código y es un perfecto ejemplo de la aplicación del lenguaje formal para compartir datos para estadísticas de morbilidad y mortalidad. Los dos siguientes códigos son un ejemplo de su asociación: código y descripción o diagnóstico.

Q392: Fístula traqueo esofágica congénita sin mención de atresia

Q750: Craneosinostosis

La utilidad del uso de la lematización es extensa desde el plano de los sistemas de información en salud. Con

* Medico Cirujano, MSc. Facultad de Salud Publica y Administración Carlos Vidal Layseca, Universidad Peruana Cayetano Heredia.

la compresión del tamaño del diccionario por la lematización se incrementa la velocidad de búsqueda y con la indización de términos comunes se reduce la ambigüedad, así cardiopatía, cardialgia, cardiaco van a ser términos relacionados por tener un común lexema. Entonces sin ambigüedad y con relaciones entre conceptos es posible buscar en forma precisa códigos relacionados en el CIE-10 y esto conllevaría a un incremento en la calidad de los documentos electrónicos en salud, llámese historias clínicas electrónicas.

Como nuestro objetivo fue desarrollar un algoritmo capaz de lematizar palabras médicas complejas, nuestros objetivos específicos fueron conceptualizar el apareamiento entre una termino de consulta y la descripción del concepto medico sin perder precisión ni redundar en nuevas búsquedas. Por otro lado, la contribución de este desarrollo no solo esta ambientado para el uso con el castellano, sino para el uso con otros lenguajes romances con raíces latinas o griegas.

Lenguaje médico

Los términos médicos difieren de las palabras corrientes, por que en ellas hay frecuentes combinaciones de muchas palabras del latín y del griego. Los antiguos lenguajes conforman alrededor del 70% de las palabras médicas en casi todos los lenguajes (como el español, ingles, francés, etc.) (4). Cada una de estas palabras en latín y griego carga un significado único y contribuye en el significado de los términos médicos. Un ejemplo de la complejidad de un término medico es:

acropaquidermoperiostosis
ACRO – PAQUI – DERMO – PERI – OSTO – OSIS

Por otro lado muchos conceptos médicos pueden ser expresados en un palabra o en varias palabras.

Por ejemplo, estos diagnósticos son sinónimos con el mismo código en el CIE-10:

1. “POLIDERMATOMIOSITIS”
2. “DERMATO-POLIMIOSITIS”
3.“ENFERMEDAD DERMICA POR POLIMIOSITIS”

Si nosotros retiramos las palabras inútiles como los artículos, preposiciones y palabras que no alteran la descripción del diagnóstico como: “enfermedad” y “por” con algoritmos convencionales producimos los siguientes conceptos claves:

1. “POLIDERMATOMIOSITI” (A)
2. “DERMAT” “POLIMIOSITI” (B)
3. “DERMIC” “POLIMIOSITI” (C)

Un simple cálculo para ver la tasa de correspondencia en el total de palabras en ambos término es:

1. A con B: 0/5 = 0%
2. A con C: 0/5 = 0%
3. B con C: 1/5 = 20%

Esto significa que solo uno de los tres pares de sinónimos produce un pobre apareamiento con todos. Sin embargo, si nosotros podríamos lematizar el término en sus componentes, nosotros tendríamos estos conceptos:

1. “POLI” “DERMAT” “MIOS” “ITIS” (A)
2. “DERMAT” “POLI” “MIOS” “ITIS” (B)
3. “DERMIC” “POLI” “MIOS” “ITIS” (C)

En este caso aplicando el mismo algoritmo para ver la tasa de correspondencia, esta mejoraría considerablemente:

A con B: 4/4 = 100%
A con C: 3/4 = 75%
B con C: 3/4 = 75%

Por esta razón, el algoritmo que buscamos debe reconocer claramente los lexemas o núcleos de significado para producir pequeños términos sin cambiar el concepto médico que tuvo originalmente.

Más sobre lematización

1. Sobre la lematización

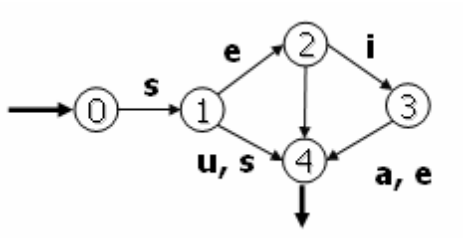
En muchos casos las variantes morfológicas de las palabras tienen interpretaciones semánticas similares y pueden ser consideradas como equivalentes para el propósito de aplicaciones para RDSI. En contraste los lenguajes naturales no son completamente regulares en sus construcciones y los clásicos lematizadores producen inevitables errores. Por otro lado las palabras que podrían ser separadas juntas (como “adherir” y “adhesión”) podrían quedar distintas después de la lematización; y las palabras que son realmente distintas podrían ser cortadas erróneamente (como “experimento” y “experiencia”) (5).

2. Revisión de investigación sobre lematización.

A. Lematizador S

Un simple método muy usado en RDSI para palabras en ingles es el lematizador S (1,2); en el cual solo unas pocas terminaciones comunes son removidas: “os”, “as” y “s” (sin excepciones). Este es conservativo y raramente reconoce en donde realizar el corte para la lematización (Figura Nº1).

Figura Nº 1: Modelo del S-lematizador.



La figura Nº1 muestra el esquema simple del S-lematizador, los números son las reglas para cada terminación. La transformación sigue estas reglas (como ies -> y, es -> e y s -> espacio vacio).

B. Lematizador de Lovins

El lematizador de Lovins (6) fue desarrollado por Jule Beth Lovins del MIT en 1968, este lematizador usa un algoritmo único y varias listas de excepciones para remover mas de 260 diferentes terminaciones. Específicamente, este simple y secuencial algoritmo (remueve un máximo de 1 sufijo por vez y contiene 11 tablas conteniendo 260 sufijos, 29 casos para remoción de sufijos y 34 reglas para decodificar terminaciones.

C. Lematizador de Porter

Es un lematizador lineal secuencial, considerado uno de los mejores y más conocidos lematizadores usados en sistemas experimentales de RDSI (7). Este remueve en cinco pasos controlados más de 60 terminaciones, removiendo terminaciones cortas sin excepciones. Cada paso resulta en la remoción de una terminación o la transformación de la raíz.

D. Lematizador de Paice-Husk

Este es un lematizador simple e iterativo que remueve las letras finales de una palabra en un indefinido número de pasos. Este algoritmo usa una lista separada de agregados finales, el cual es un arreglo de una lista que esta dividida en series y secciones, cada una de las cuales es correspondiente a una letra del alfabeto.

E. Otras técnicas

1. Lematizador de Krovetz: Krovetz introdujo un diccionario flexible y declarativo junto con un nuevo y moderado lematizador derivacional (8).

2. Lematizador Nice: Este es una combinación del lematizador original de la idea de la agregación de los lematizadores S-stemmer, Porter y Krovetz. Su algoritmo sirve al usuario en una variedad de necesidades; este sistema acepta 3 formas de entradas: palabras, archivos e índices de archivos (9).

Los componentes y reglas en palabras médicas

La palabra es la unidad mínima de significado; esta puede tener una combinación de raíces, prefijos, sufijos y letras finales. Un término médico se compone de la combinación de ellas. Pero para cuestiones de significado solo algunas partes aportan significado y otras sirven para dar atributos o sostén a la descripción del concepto.

1. Los componentes válidos de un término médico.
Los sufijos son a menudo usados para plural, y atributos del verbo, los que podrían ser removidos. Sin embargo, especialmente in las palabras médicas estos sufijos cargan un importante significado, como por ejemplo el sufijo "–ITIS" que significa una inflamación o el sufijo "–OSIS" que significa un proceso degenerativo. Ellos tienen que ser preservados por que cargan un significado importante para la palabra médica, pero tienen que ser cortados en orden a conseguir la verdadera raíz de la palabra. Lo remanente luego de remover sufijos y lexemas generalmente es frecuentemente una combinación de prefijos como "A-", "EPI-", "HEMI-", "POLI-", y raíces como "NEURO-" o "CEFALO-". Ambos siempre cargan información importante y por supuesto tienen que se preservados.

2. Los componentes inválidos de un término médico.
Estos componentes inválidos son las uniones y eufónimos. Cuando las raíces están juntas en una palabra medica, estas raíces esta juntas usualmente por uniones conectores como "–e-", "-i-", "-o-", "-eo-", "-io-", "-ico-", o "–ato-". Generalmente, las palabras finales son llamadas eufónimos por que ellos no agregan significado a la palabra, en vez de eso su función es dar un buen sonido a la palabra. Una raíz puede tener diferentes conectores y eufónimos dependiendo de la siguiente raíz, sufijo como por ejemplo:

"cardialgia" : "CARD" "-I-" "ALGIA"
"cardiopatía": "CARD" "-IO-" "PATIA"
"cardiaco": "CARD" "-I-"ACO"

Los conectores no agregan mas significado y pueden ser descartados. Sin embargo es necesario ser muy cuidadosos en esto porque hay algunas raíces que tienen como letras finales, letras que son importantes y que lo diferencian de otra raíz y tiene un significado diferente en el sentido de distinguirlos de otra raíz, como por ejemplo "HEMI-" y "HEMO-" o "ANTI-" y "ANTE-" (Figura Nº2).

Figura Nº 2: Modelo de la composición de una palabra medica compleja usadas en nuestro sistema.

Prefijo	L	Raiz 1	L	…	Raiz n	Sufijo	Eufonio

3. Reglas generales en el proceso de lematizar.

En el sentido de identificar raíces, prefijos, sufijos, conectores y eufónimos en las palabras médicas en forma correcta, nosotros establecimos reglas para cada una de estas clases. Estas reglas contenían la forma normalizada de un lematizador con prefijos y sufijos sin sus usuales conectores o eufónimos.

4. Las reglas especificas para los componentes cortos. Algunas veces, los componentes cortos que son usualmente los prefijos son a menudo falsamente reconocidos, como por ejemplo el prefijo A- que esta presente en la palabra afaquia, podría ser falsamente removido en la palabra arteria destruyendo su significado:
"arteria" quitándole seria "A" – "RTERIA"

Algunas veces este proceso es bueno:
"afonía" produciendo dos conceptos "A"- "FONIA"

Este cuidado previene que futuras raíces de un término puedan eliminarse en forma errada y puedan formar una descomposición caótica de los restos de la palabra. Este problema fue encontrado con suma importancia en las palabras con menos de 3 letras. La regla que se asumió para esto es más o menos así:

Código LISP
… (COND ((AND (= 0 (SEARCH X STR))(STRING-EQUAL X "A")(FIND (SUBSEQ STR 1 4) A-EXCEPTIONS)) T) …

Esto significa
Un componente con menos de 3 letras (**x**) es solo separado de la raíz (str), si esta es seguida por un componente conocido dentro de las excepciones para la letra "a" (a-excepciones).

METODOLOGÍA

Luego de conceptualizado el problema con los conceptos médicos se trabajo en lo siguiente:

Creación del algoritmo de lematización: Para lo cual se eligió un lenguaje declarativo de trabajo y de fácil adaptación a listas, este fue el Lenguaje LISP. Como software para declarar el algoritmo se uso el Allegro Common Lisp versión 6.2 para Windows.

Creación de las listas y reglas: Estas listas fueron creadas usando un diccionario de latín y griego, estas listas fueron escritas en formato texto.

La declaración del algoritmo empieza por tres pasos fundamentales:

En el primer paso, una lista determina la presencia de un prefijo, raíz o sufijo conocido en la palabra.

El segundo paso ocurre solo si la palabra no tiene ningún prefijo, raíz o sufijo conocido. En el segundo paso, un algoritmo que lematiza en forma estándar remueve los plurales, atributos de los verbos o eufónimos.

El tercer y último paso es descomponer la palabra en raíces, prefijos y sufijos de acuerdo con las reglas preestablecidas (Figura Nº3).

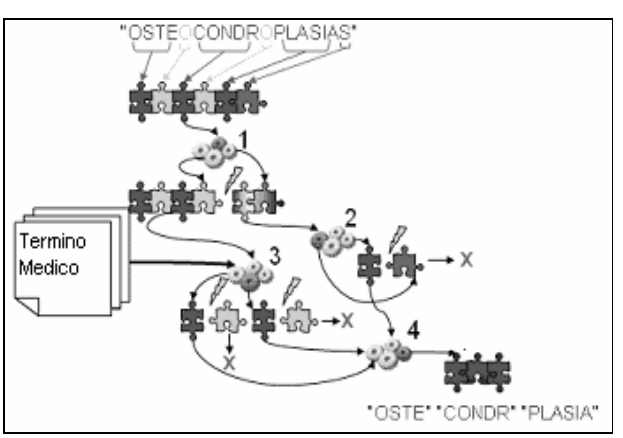
La figura Nº3 muestra los 4 sub. pasos de los 3 pasos. El sub-paso 1 guarda el sufijo y el eufónimos (letra final) por que no existe un conector entre estas partes; el sub.-paso 2 descompone el conector entre el sufijo y el eufónimos de acuerdo a las reglas; el sub-paso 3 termina de descomponer el resto de las palabras de acuerdo a las reglas en las palabras médicas y borra los conectores. El sub-paso 4 une los posibles prefijos, raíces y sufijos en un solo texto.

RESULTADOS

1. Implementación

El algoritmo fue implementado en LISP. Este es un lenguaje de programación muy usado en sistemas

Figura Nº 3: Pasos de la descomposición de las palabras médicas complejas con nuestro sistema.



expertos y para Inteligencia Artificial, además trabaja muy bien con listas y texto.

El algoritmo es secuencial y con moderación en cada uno de ellos, es decir si uno de los pasos no es completado se suspende el proceso y se devuelve la palabra sin modificación, esta secuencias es de tres etapas:

Para el paso uno nosotros desarrollamos un algoritmo que trabaja consultando un diccionario de sufijos y prefijos.

El Segundo algoritmo es similar al algoritmo de Porter con la diferencia que esta adaptado a palabras médicas en castellano.

La separación de los componentes en el paso tres es realizado bidireccionalmente desde el inicio hasta el fin de la descomposición de la palabra. Esto asegura que si el término contiene un componente desconocido, un máximo de componentes conocidos son reconocidos. Por otro lado, las listas de las reglas y el componente son ordenados por longitud haciendo seguro que siempre se busque primero la parte mas larga de los componentes.

Finalmente, nosotros evaluamos el algoritmo modular con una lista de más de 15,000 términos médicos que fueron extraídos del CIE-10 (3). Todos los términos fueron descompuestos en sus lexemas y se guardo firmas conceptuales para cada término específico.

Por ejemplo:
1. El algoritmo arregló términos complejos con muchos prefijos y sufijos en forma correcta:
“Hipergammaglobulinemias”: “HIPER” – “GAMMA” –”GLOBUL” – “EMIA”

2. También nombres y formas de atributos son arreglados para un mismo concepto:

“leucocito”: “LEUCO” – “CIT”
“leucocitarios”: “LEUCO” – “CIT”
“displasia”: “DIS” – “PLAS”
“displasicas”: “DIS” – “PLAS”

3. Y los prefijos, raíces y sufijos mantuvieron su propia firma conceptual:

“afasia”: “A” – “FASI”
“anhidrosis”: “AN” – “HIDR” – “OSIS”
“anaerobio”: “AN” – “AERO” – “BIO”
“antebrazo”: “ANTE” – “BRAZO”
“anterior”: “ANTERIOR”

2. Limitaciones

Los resultados presentan una buena performance y efectividad pero estos dependen principalmente de disponer de una lista completa de términos médicos. Estas listas que deben ser preparadas exhaustivamente necesitan muchas veces realizarse manualmente y allí se pueden inducir errores.

DISCUSION

El algoritmo desarrollado para lematizar términos médicos complejos abre un espacio de investigación en la aplicación de los sistemas de información en los registros médicos. Si bien es cierto que esta es un área para las escuelas de informática y computación ya lo es para los nuevos programas de informática en salud. En consecuencia, el desarrollo de motores de búsqueda comprensivos y potentes para los registros médicos no solo está basado en el avance de los procesadores sino también en los algoritmos de búsqueda. Nuestra contribución busca llenar ese espacio de investigación operativa de la aplicación de los sistemas de información sobre el lenguaje medico. Una aplicación cercana de este algoritmo es la codificación automática de vocabularios controlados como CIE-10, con la misma idea no es difícil plantear su integración para el uso de sistemas expertos de diagnósticos con el fin de reconocer palabras agrupadas por comunes lexemas o procesos comunes.

El alcance de este trabajo va principalmente para la nueva área de informática médica sobre todo para el uso de sistemas expertos para la búsqueda de conceptos médicos y lenguaje médico estructurado.

Finalmente, como futura investigación, nosotros planteamos que su implementación en el mundo real debe ser aplicada en historias clínicas electrónicas para la búsqueda de conceptos médicos afines.

Agradecimientos:

Al Dr. Ira Kalet de la Universidad de Washington por sus valiosas sugerencias con el lenguaje LISP.

Correspondencia:

Jesús Peinado. 2708 54th NE
Seattle – WA. 98105 USA.
Tel. (206)523-0306.
jpeinado@u.washington.edu

REFERENCIAS BIBLIOGRAFICAS

1. Hull D and Grefenstette G. "A detailed analysis of English stemming algorithms", Rank Xerox Research Centre, January 31, 1996.
2. Harman D. "How effective is suffixing?". Journal of the American Society for Information Science 1991;42(1): 7-15.
3. WHO, Classification Statistical International of Diseases tenth edition in Spanish. Geneva -1998, Vol I, II and III.
4. Nascimento Mario, da Cumba A. An experiment Stemming Non-Traditional text www.dcc.unicamp.br/-mario.
5. Fox B, J. Fox C. Efficient stemmer generation. Information Processing and Manegement 2002; 38: 547-558.
6. Lovins J,B. "Development of a stemming algorithm, Mechanical Translation and Computational Linguistics".1968; 11(1-2): 22-31.
7. Porter M. "An algorithm for suffix stripping, 1980;Program 14(3): 130-137.
8. R. Krovetz, Viewing morphology as an inference process, in : R. Korfhage, E Rasmussen, P. Willet (Eds), "Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval", ACM, Pittsburg, USA, 1993:191-202.
9. Xu J, Croft B. Corpus-Based Stemming using Co-occurrence of word variants. Computer Science Department. University of Massachusetss, Ambersy.