Sistemas de detección y extracción semiautomática de siglas

Estado de la cuestión

John Jairo Giraldo Ortiz

Índice

Resumen *3*

Introducción *4*

I. Sistemas de detección y extracción de pares sigla-FD *6*

A. Métodos basados en patrones **7**

1. Acronym Finder Program (AFP) **7**

2. Three Letter Acronym (TLA) **11**

3. Acrophile **13**

4. Acromed **21**

5. Sistema para gestión de variación terminológica **25**

6. A simple algorithm **29**

B. Métodos basados en estadística y aprendizaje máquina **32**

1. Métodos basados en técnicas estadísticas **32**

1.1. Diccionario de abreviaciones en línea **32**

1.2. A Simple and Robust Abbreviation Dictionary (SaRAD) **35**

2. Métodos basados en algoritmos de aprendizaje máquina **41**

2.1 Teoría universal de la formación de siglas **41**

2.2 Automatic Acronym Identification and Creation of an Acronym Database **46**

2.3 A supervised learning approach **50**

2.4 Recognizing acronyms in Swedish texts **52**

3. Métodos híbridos **56**

II. Sistemas de desambiguación de siglas *61*

A. Polyfind **62**

B. Automatic resolution of ambiguous abbreviations in Biomedical texts **63**

III. Criterios para el diseño de un modelo de detector-extractor de siglas para el español *65*

Conclusiones *72*

Bibliografía *74*

Resumen

En la actualidad existen diversos diccionarios de siglas en línea. Entre ellos sobresalen Acronym Finder, Abbreviations.com y Acronyma; todos ellos dedicados mayoritariamente a las siglas inglesas. Al igual que los diccionarios en papel, este tipo de diccionarios presenta problemas de desactualización por la gran cantidad de siglas que se crean a diario. Por ejemplo, en 2001, un estudio de Pustejovsky et al. mostraba que en los abstracts de Medline aparecían mensualmente cerca de 12.000 nuevas siglas. El mecanismo de actualización empleado por estos recursos es la remisión de nuevas siglas por parte de los usuarios. Sin embargo, esta técnica tiene la desventaja de que la edición de la información es muy lenta y costosa. Un ejemplo de ello es el caso de Abbreviations.com que en octubre de 2006 tenía alrededor de 100.000 siglas pendientes de edición e incorporación definitiva. Como solución a este tipo de problema, se plantea el diseño de sistemas de detección y extracción automática de siglas a partir de corpus. El proceso de detección comporta dos pasos; el primero, consiste en la identificación de las siglas dentro de un corpus y, el segundo, la desambiguación, es decir, la selección de la forma desarrollada apropiada de una sigla en un contexto dado. En la actualidad, los sistemas de detección de siglas emplean métodos basados en patrones, estadística, aprendizaje máquina, o combinaciones de ellos. En este estudio se analizan los principales sistemas de detección y desambiguación de siglas y los métodos que emplean. Cada uno se evalúa desde el punto de vista del rendimiento, medido en términos de precisión (porcentaje de siglas correctas con respecto al número total de siglas extraídas por el sistema) y exhaustividad (porcentaje de siglas correctas identificadas por el sistema con respecto al número total de siglas existente en el corpus). Como resultado, se presentan los criterios para el diseño de un futuro sistema de detección de siglas en español.

Resum

Actualment existeixen diversos diccionaris de sigles en línia. Entre ells destaquen l'Acronym Finder, l'Abbreviations.com i l'Acronyma; tots ells orientats majoritàriament a recollir les sigles angleses. Tal com succeeix amb els diccionaris en paper, aquest tipus de diccionaris presenta problemes de desactualització per la gran quantitat de sigles que es creen diàriament. Per exemple, al 2001, un estudi de Pustejovsky et al. mostrava que en els abstracts de Medline apareixien mensualment prop de 12.000 noves sigles. El mecanisme d'actualització emprat per aquests recursos és la remissió de noves sigles per part dels usuaris. No obstant això, aquesta tècnica té el desavantatge que l'edició de la informació és molt lenta i costosa. Un exemple d'això és el cas de l'Abbreviations.com, que a l'octubre de 2006 tenia al voltant de 100.000 sigles pendents d'edició i incorporació definitiva. Com a solució a aquest tipus de problema, es planteja el disseny de sistemes de detecció i extracció automàtica de sigles a partir de corpus. El procés de detecció comporta dos etapes: la identificació de les sigles dintre d'un corpus i la desambiguació, és a dir, la selecció de la forma desenvolupada apropiada d'una sigla en un context determinat. En l'actualitat, els sistemes de detecció de sigles usen mètodes basats en patrons, estadística, aprenentatge màquina, o combinacions d'ells. En aquest estudi s'analitzen els principals sistemes de detecció i desambiguació de sigles i els mètodes que aquests utilitzen. Cadascun d'ells s'avalua des del punt de vista del rendiment, mesurat en termes de precisió (percentatge de sigles correctes pel que fa al nombre total de sigles extretes pel sistema) i exhaustivitat (percentatge de sigles correctes identificades pel sistema pel que fa al nombre total de sigles existent en el corpus). Com a resultat, es presenten els criteris per al disseny d'un futur sistema de detecció de sigles en espanyol.

Abstract

At present, there are several online acronym dictionaries, among which Acronym Finder, Abbreviations.com and Acronyma are the most important ones. All of them have been developed mainly to deal with English acronyms. Like paper dictionaries, online dictionaries have the problem of being updated because of the high amount of acronyms created everyday; e.g., in 2001, a study carried out by Pustejovsky et al. showed that in Medline abstracts about 12.000 new acronyms appeared monthly. The update mechanism used by these dictionaries consists in the acquisition of new acronyms through users. However, the main disadvantage of this mechanism is that information editing is very expensive and slow; e.g., in October 2006, Abbreviations.com had about 100.000 acronyms to edit and store definitely. To solve this problem, the design of automatic acronym recognition and disambiguation systems from corpus has been proposed. The recognition process implies two steps, namely the identification of acronyms from corpus, and the selection of the proper acronym expansion from a given context. Currently, the recognition systems use methods based on patterns, statistics, machine learning or even a combination of them. In this work, the main acronym recognition and disambiguation systems are analysed. Each of them are evaluated from a performance-oriented point of view. Thus, precision, i.e. the percentage of correct acronyms with regard to the total number of acronym recognised by the system is measured as well as recall; i.e., the percentage of correct acronyms recognised by the system with regard to the total number of acronyms in the corpus. As a result, a set of criteria is presented in order to outline a future acronym identification system for the Spanish language.

Introducción

Hoy en día dominios como la informática, las telecomunicaciones, la biología molecular y la genética presentan una rápida evolución, la cual se refleja en la generación de conceptos y denominaciones nuevos.

En los textos especializados muchas denominaciones suelen acortarse mediante procesos como la siglación para ajustarse a criterios estilísticos o editoriales. Normalmente, una sigla va acompañada de su forma desarrollada (FD) la primera vez que aparece dentro de un documento. A partir de allí, suele aparecer sola; lo cual puede dificultar la comprensión a aquellos lectores que no son expertos en el tema.

Existen muchos diccionarios de siglas de carácter especializado, tanto en formato papel como electrónico. Los diccionarios en papel no pueden actualizarse automáticamente, por lo que, inevitablemente, llegan desactualizados al público. Autores como Gehénot (1990: 105) han investigado sobre la producción de este tipo de recursos a lo largo de la historia. En su estudio destaca obras como el *Tractatus de Siglis Veterum* (1703);[1] *Abréviations de sociétés, conventionnelles et usuelles* (1926),[2] o el *Dictionnaire d'abréviations françaises et étrangères, techniques et usuelles, anciennes et nouvelles* (1951).[3]

Como se indicaba antes, el auge de la ciencia y la tecnología ha sido un factor clave para que el número de recursos sobre siglas continúe en aumento. En las últimas décadas han aparecido obras como: *Dictionnaire international d'abréviations scientifiques et techniques* (1978), *Diccionario internacional de siglas y acrónimos* (1984), *Dictionnaire des abréviations et acronymes scientifiques, techniques, médicaux, économiques et juridiques* (1992), *Elsevier's Dictionary of Acronyms, Initialisms, Abbreviations and Symbols* (1997), y *Acronyms, Initialisms and Abbreviations Dictionary*, 32ª edición (2003).

[1] Esta obra escrita en latín, comprende 314 páginas y 49 capítulos que estudian en detalle el uso de las siglas en un tema particular: derecho, medicina, aritmética, gramática, música, numismática, etc.
[2] Abreviaturas marítimas, bursátiles, comerciales en francés, inglés, alemán y español etc.
[3] Este diccionario reúne 8.000 abreviaturas de artes, automoción, aviación, banca, cartografía, química, comercio, derecho, electricidad, finanzas, impuestos, industria, jurisprudencia, marina, matemáticas, mecánica, medicina, etc.

Los diccionarios electrónicos de siglas, también denominados diccionarios *on line*, surgieron a finales de los años ochenta. Funcionan como bases de datos consultables, cuyo objetivo es responder a la necesidad de conocer las FD de la gran cantidad de siglas que se producen en todos los campos de conocimiento. Su ventaja respecto de los diccionarios en papel, radica en que permiten almacenar y actualizar rápidamente la información.

Existen diversos diccionarios de abreviaciones (principalmente siglas) disponibles en internet entre los que sobresalen *Acronym Server* (1988), *Acronym Finder* (1996), *Wiley InterScience* (1999), *Abbreviations.com* (2001), y *Acronyma* (2004). Generalmente, esta clase de diccionarios funciona mediante una interfaz de consulta donde se puede buscar dos tipos de información: una FD o una palabra específica dentro de todas las FD existentes en el diccionario.

A pesar de lo anterior, este tipo de diccionarios tampoco es ajeno a la desactualización, inevitable a causa de la gran cantidad de siglas que se crean a diario.[4] La puesta al día de recursos como *Acronym Finder*, *Abbreviations.com* o *Acronyma* depende del envío de nuevas siglas por parte de los usuarios, las cuales se someten primero a un proceso de edición, que puede tardar varias semanas.[5]

Este informe presenta las generalizaciones sobre los sistemas de detección y desambiguación de siglas así como los parámetros que debería aplicar un sistema de detección de siglas para el español.

[4] En 2001, un estudio de Pustejovsky *et al.* mostró que cerca de 12.000 siglas se creaban mensualmente en los *abstracts* de *Medline*.
[5] A 19 de octubre de 2006, Abbreviations.com tenía cerca de 100.000 siglas por editar e incorporar definitivamente.

I. Sistemas de detección y extracción de pares sigla-FD

Desde finales de la década de los noventa han surgido diversos sistemas para el tratamiento automático de siglas. La creación de estos sistemas ha llevado a los investigadores a buscar paralelamente soluciones para dos tipos de problemas: la detección y la desambiguación.

Se denomina detección y extracción al proceso mediante el cual las siglas se recopilan, manual o automáticamente, a partir de corpus textuales.[6] Se denomina desambiguación al mecanismo mediante el cual se selecciona la FD apropiada de una sigla en un contexto dado.

Durante la última década se ha dado un desarrollo vertiginoso en lo que se refiere a los sistemas de detección y extracción de siglas. Este periodo se ha destacado por dos hechos; en primer lugar, el fomento de la investigación en este campo por parte de la biomedicina y la informática. Y, en segundo lugar, el predominio del inglés como lengua objeto de estas investigaciones.

Los sistemas de detección y extracción de siglas actuales emplean métodos basados en patrones, estadística, aprendizaje máquina, o en una combinación de éstos (híbridos).

En este apartado se describen estos métodos y los principales sistemas que los emplean. Dentro de cada sistema se analizan los patrones y la técnica de detección, así como la evaluación del rendimiento, medida en términos de precisión y exhaustividad.

La precisión consiste en medir el porcentaje de siglas correctas con respecto al número total de siglas extraídas por el sistema, mientras que la exhaustividad mide el porcentaje de siglas correctas identificadas por el sistema con respecto al número total de siglas existente en el corpus.

[6] A este proceso también se le conoce como identificación, reconocimiento o adquisición.

A. Métodos basados en patrones

Los métodos más tradicionales para hallar pares sigla-FD se basan en la coincidencia de patrones. Estos métodos difieren unos de otros en cuanto al tipo de información que codifican en sus reglas, las cuales son cruciales en el rendimiento del sistema.

Entre los sistemas que emplean métodos basados en patrones se destacan: *Acronym Finder Program*, *Three Letter Acronym*, *Acrophile*, *Acromed*, *A simple algorithm* y *Sistema para la variación terminológica.*

1. Acronym Finder Program (AFP)

Acronym Finder Program (Taghva & Gilbreth, 1999) es una herramienta que usa el algoritmo *Longest Common Subsequence* (LCS) para hallar todas las alineaciones posibles entre un candidato a sigla y su FD. AFP se evaluó en un corpus de documentos oficiales sobre el Proyecto de disposición de basuras de *Yucca Mountain*.

El trabajo de Taghva & Gilbreth es de gran importancia por cuanto es el pionero en el estudio de los sistemas de detección y extracción de siglas; de ahí que sea referencia frecuente en todas las publicaciones del área.

a. Patrones

- Todas las palabras mayúsculas, desde tres hasta diez caracteres, son aceptadas como candidatos a sigla
- Cada caracter de la sigla debe coincidir con el primer caracter de cada palabra de la FD.

b. Técnica de extracción de siglas

El proceso de extracción de siglas de AFP consta de cuatro fases: inicialización, filtro, *parser* y aplicación del algoritmo.

1) Inicialización

La entrada de datos para el algoritmo consta de los siguientes componentes:

- *Palabras vacías o stopwords*. Consiste en una lista de las palabras que generalmente se omiten en la formación de siglas; es decir, artículos, preposiciones y conjunciones.
- *Palabras rechazadas*. Consiste en una lista opcional de palabras frecuentes en los documentos, y que se sabe que no son siglas; por ejemplo: TABLE, FIGURE, números romanos, etc.).
- *Base de datos de siglas*. Esta información puede ser usada para ignorar la rutina de búsqueda del programa o para repetir el proceso cuando la búsqueda no arroje resultados. Se trata de un mecanismo opcional.
- *Corpus.* Consiste en el texto o conjunto de textos a rastrear.

2) Filtro de datos

La entrada de datos se procesa para descartar líneas de texto en mayúsculas como pueden ser los títulos. Cuando el programa identifica un candidato a sigla consulta la lista de palabras rechazadas. Si el candidato no aparece en dicha lista, entonces el proceso continúa con la búsqueda de su FD en el texto que lo rodea. Para ello, el programa crea una ventana de texto formada por dos subventanas llamadas ventana anterior y ventana posterior. La longitud de cada ventana (medida en palabras) se establece al multiplicar por dos el número de caracteres de la sigla.

3) Parser

El sistema prioriza diferentes tipos de palabras para que el algoritmo encuentre un número razonable de FD, así:

- *Palabras vacías*. No pueden eliminarse del proceso de búsqueda de las FD. Si el algoritmo ignora por completo este tipo de palabras, muchas siglas pueden pasarse por alto. Aunque el sistema da prioridad a los elementos que no son palabras vacías, Taghva & Gilbreth reconocen que hay casos en los que estas deben tenerse en

cuenta; *e.g.*: *Department of Energy (DOE).* Pero, por el contrario, hay ocasiones en que las palabras vacías deben ignorarse; *e.g.*: *Office of Nuclear Waste Isolation (OWNI).*

- *Palabras separadas por guiones*. Las FD contienen a menudo palabras separadas por guiones. En este sentido, puede darse uno de los siguientes casos:
  - que la primera palabra separada por guión pertenezca a la FD; *e.g.*: *X-ray photoelectron spectroscopy (XPS).*
  - que todas las partes de la unidad separada por guión pertenezcan a la FD; *e.g.*: *non-high-level solid waste (NHLSW).*
- *Siglas*. Dentro de los textos, las siglas pueden estar cerca unas de otras como cuando las siglas incluyen otras siglas en sus formas desarrolladas; *e.g.*: *ARINC Communications and Reporting System (ACARS).*
- *Palabras que no pertenecen a ninguno de los tipos anteriores*. Esta clase de unidades constituye la mayor parte de las palabras de las FD y no necesita de un tratamiento especial durante el proceso.

Cuando se aplica el *parser* a una subventana, se generan dos patrones de símbolos. El primer patrón se denomina *líder* (o caracter inicial de cada palabra) y el *tipo* (o clase de palabra presente en la subventana). Los tipos de palabras se representan así:

s = *stopword*
H = parte inicial de una palabra separada por guión
h = partes contiguas a la palabra separada por guión
a = sigla
w = palabra normal

Por ejemplo, dado el texto:

[…] spent fuel and recycling the recovered uranium and plutonium results in the generation of transuranic (TRU) non-high-level solid waste (NHLSW). Volumes and characteristics of these wastes, and methods for […]

La ventana anterior para la sigla NHLSW es:

[results in the generation of transuranic (TRU) non-high-level solid waste]

Los patrones *líder* y *tipo* son:

[r	i	t	g	o	t	t	n	h	l	s	w]
líderes											
[w	s	s	w	s	w	a	H	h	h	w	w]
tipo											

4) Aplicación del algoritmo

Para encontrar un candidato a FD, el algoritmo identifica una subsecuencia común de letras de la sigla y del patrón *líder*. Una *subsecuencia* es justo una secuencia dada con algunos elementos removidos. Para las secuencias *X* y *Y*, decimos que una secuencia *Z* es una *subsecuencia común* de *X* y *Y* si es una subsecuencia tanto de *X* como de *Y*; *e.g.*:

Si *X*= *acbceac* y *Y*= *cebaca*, entonces *cba* es una subsecuencia común de *X* y *Y* de longitud 3.

Obsérvese que *ceac* y *cbca* también son subsecuencias comunes de *X* y *Y* (longitud 4). Obsérvese además que no hay subsecuencias comunes mayores a longitud 4; es decir, *ceac* es una subsecuencia común de máxima longitud. La subsecuencia común más larga (LCS) de cualquiera de las dos cadenas *X* y *Y* es una subsecuencia común con la mayor longitud entre todas las subsecuencias comunes.

c. Evaluación

Para evaluar AFP se tomó un corpus de 17 documentos. El sistema identificó correctamente 398 siglas, que en términos de rendimiento implica 98% de precisión y 86% de exhaustividad.

A partir de los resultados anteriores, los autores excluyeron las siglas menores o iguales a dos caracteres. De esta manera, lograron aumentar la exhaustividad hasta 93%, mientras que la precisión se mantuvo en el mismo nivel, es decir, en 98%.

AFP sólo considera como candidatos a sigla las cadenas de tres o más caracteres en mayúscula, dejando por fuera un gran número de candidatos que pueden estar formados por sólo dos caracteres. Adicionalmente, si no se da una correspondencia exacta entre los caracteres de la sigla y las iniciales de cada una de las palabras de la FD; *e.g.*: *Teledyne Wachang Albany* (TWCA), el sistema será incapaz de reconocerlo como un candidato válido.

2. Three Letter Acronym (TLA)

TLA es un sistema creado por Stuart Yeates (1999) para la detección y extracción de pares sigla-FD a partir de documentos de la Biblioteca digital de Nueva Zelanda.

a. Patrones

- Las siglas son más cortas que sus FD
- Las siglas contienen las iniciales de la mayoría de las palabras de sus FD
- Las siglas se forman con letras mayúsculas
- Las siglas más cortas tienden a tener palabras más largas en sus FD
- Las siglas más largas tienden a tener más *palabras vacías.*

b. Técnica de extracción de siglas

Inicialmente, un analizador léxico toma una secuencia de texto sin procesar, de la que selecciona los candidatos a sigla y sus FD. Luego, estos candidatos se pasan por un revisor heurístico, el cual aplica un número de reglas para descartar las concordancias falsas provenientes del analizador léxico. Posteriormente, en la fase de refinamiento, se eliminan los duplicados de las siglas resultantes. La figura 1 muestra la estructura general del sistema.

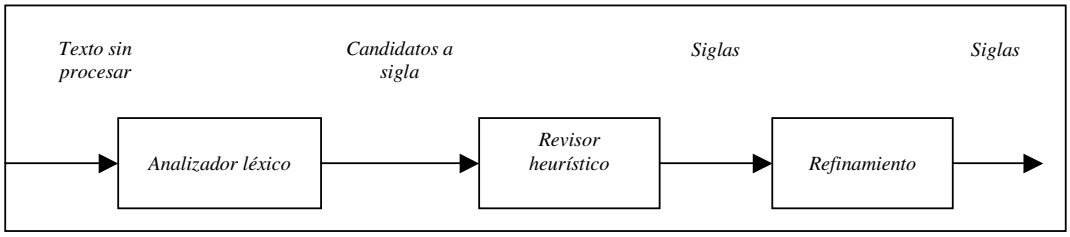

Fig. 1. Estructura general del extractor de siglas (Yeates, 1999)

El analizador léxico ejecuta dos funciones. De un lado, remueve todos los caracteres que no son alfabéticos y divide el texto en trozos *"chunks"* basándose en la ocurrencia de los caracteres de la coma (,) y el punto (.), los cuales indican el final de un *chunk* y el comienzo de otro; por ejemplo:

el texto: **Ab cde (fgh ijk) lmn o p. Qrs**
se divide en: **Ab cde | fgh ijk | lmn o p | Qrs**

Posteriormente, en cada *chunk* se tiene en cuenta cada palabra para determinar si es un candidato a sigla. Se compara con los *chunks* anterior y posterior para buscar una FD concordante. De esta manera se generan los siguientes pares:

Ab fgh ijk
cde fgh ijk
fgh Ab cde
ijk Ab cde
fgh lmn o p
ijk lmn o p
... ...

Si se encuentra un candidato a FD, el par sigla-FD se convierte en un candidato y se pasa por el revisor heurístico.

El analizador léxico usa un algoritmo al momento de buscar las FD, como puede apreciarse en la figura 2.

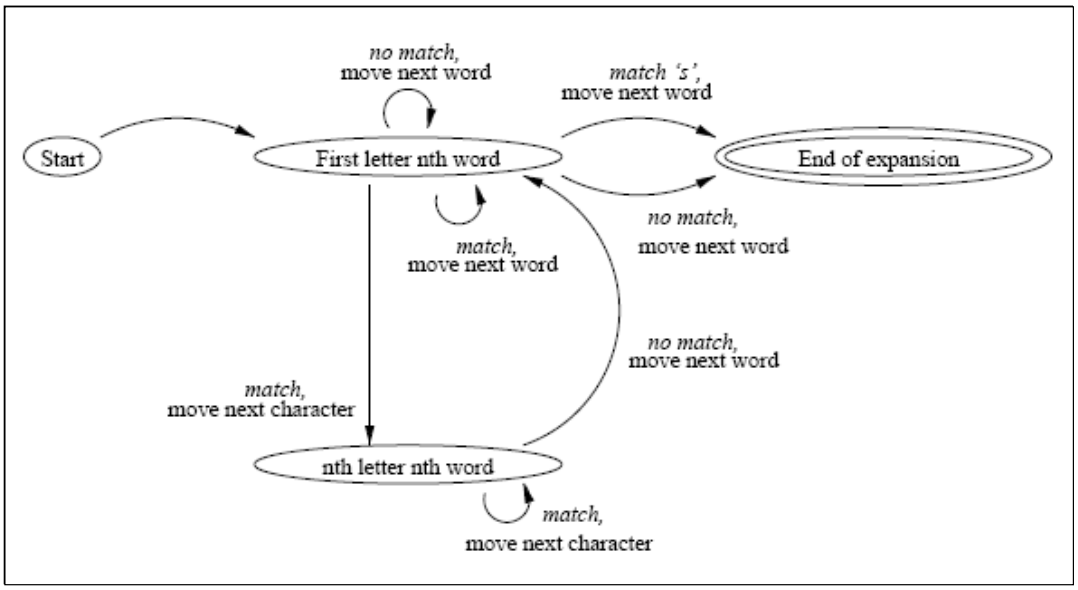

Fig. 2. Algoritmo de concordancia de siglas (Yeates, 1999)

c. Evaluación

El sistema TLA no logra reconocer palabras dentro de la FD que contienen más de una letra mayúscula como *DBMS* (*DataBase Management System*).

El corpus de evaluación de TLA constaba de 10 informes técnicos de informática. El rendimiento alcanzado fue de 68% de precisión y 91% de exhaustividad.

3. Acrophile

Larkey *et al.* (2000) desarrollaron *Acrophile*, un sistema que contiene una colección de siglas y sus formas desarrolladas recogidas de gran cantidad de páginas web mediante un proceso heurístico de extracción. Para llevar a cabo este proyecto se evaluaron y compararon cuatro algoritmos.

Acrophile permite a los usuarios hacer dos tipos de consultas. Por un lado, se puede buscar las FD correspondientes a una sigla como *IRS*, y, por otro lado, se puede buscar la sigla correspondiente a una FD como *Internal Revenue*. El sistema produce listas de siglas y FD clasificadas por puntaje de calidad.

Como prestación adicional, este sistema ofrece una casilla para introducir direcciones de páginas web, las cuales son rastreadas con el fin de encontrar pares de candidatos sigla-FD.

a. Patrones

Los patrones para la identificación de siglas se detallan en el apartado 1 "Detección de siglas" y en la tabla 1 "Características de las siglas y sus FD".

b. Técnica de extracción de siglas

Como se dijo anteriormente, los algoritmos de *Acrophile* utilizan el analizador léxico *flex* y el *parser yacc* para procesar textos y extraer sus siglas. Las FD se detectan en el texto por medio de una combinación de reglas contextuales y canónicas, las cuales coinciden con los patrones con que se expresan comúnmente las FD en el inglés estándar escrito.

Larkey *et al.* implementaron y probaron cuatro algoritmos de extracción diferentes. Todas las versiones trabajan bajo el principio general de que una secuencia de caracteres es una sigla si cumple con ciertos patrones, confirmándola como sigla cuando se encuentra cerca una FD coincidente.

Después de realizada la extracción, se lleva a cabo un proceso de normalización para todos los algoritmos: dos siglas se consideran equivalentes si sólo difieren en el uso de mayúsculas o en la presencia o ausencia de puntos, guiones o espacios. Dos FD de una sigla se consideran equivalentes si sólo difieren en el uso de mayúsculas.

Los cuatro algoritmos, denominados *contextual*, *canónico*, *canónico-contextual* y *simple* difieren en cuanto a:

- los patrones que toman para indicar los candidatos a sigla,
- los tipos de FD que pueden encontrarse, y
- los patrones de texto que indican un candidato par sigla-FD.

Los algoritmos contextual, canónico y canónico-contextual están relacionados y surgen de la modificación de un algoritmo contextual previo. El algoritmo simple se desarrolló de forma independiente para probar un enfoque más limitado y lograr mayor precisión en los candidatos a sigla. Cada algoritmo se probó y perfeccionó mediante la aplicación de una prueba piloto que consistió en procesar un corpus de 12.380 artículos del *Wall Street Journal*.

El algoritmo simple es el más riguroso; sólo busca aquellos pares sigla-FD que se ajusten a un pequeño grupo de formas canónicas tales como: "forma desarrollada (SIGLA)" o "SIGLA or forma desarrollada".

El algoritmo contextual, mucho menos estricto, busca una FD en el contexto de la sigla sin requerir ningún patrón canónico (o, paréntesis, comas, etc.) que indique su relación.

Los algoritmos canónico-contextual y canónico se encuentran a medio camino entre los dos anteriores. En la tabla 1 pueden observarse las características de los cuatro algoritmos así como las características de las siglas y las FD.

1) Detección de siglas

Para la identificación de los candidatos a sigla, los algoritmos buscan los patrones indicados en la primera fila de la tabla 1. Esta fila presenta una notación de expresión seudo-regular en la que:

- + indica una o más ocurrencias de un símbolo
- indica 0 ó más ocurrencias
- los *superscripts* numerados indican un número específico o rango de ocurrencias
- U significa una letra mayúscula
- L significa una letra minúscula
- D significa un dígito
- S significa un final opcional con *s* o *'s*
- {sep} es un punto o punto seguido por un espacio, y

- {dig} es un número entre 1 y 9, opcionalmente seguido de un guión. Los términos en corchetes son alternativos.

Seguidamente se indican los patrones que reconoce cada uno de los algoritmos:

- *Algoritmo contextual*. Acepta siglas que tienen:
  - mayúsculas sostenidas como *USA*
  - puntos como *U.S.A.*
  - una secuencia de letras minúsculas, cuyos caracteres pueden aparecer al final del patrón siguiendo al menos tres caracteres en mayúscula, como *COGSNet*, o internamente siguiendo al menos dos caracteres en mayúscula, como *AChemS*
  - un patrón de mayúsculas también puede tener cualquier número de dígitos en cualquier posición.
- *Algoritmos canónico-contextual y canónico*. Aceptan una amplia gama de patrones de siglas. Tienen menos restricciones respecto de las secuencias de letras minúsculas para permitir patrones como *DoD.* Permiten barras y guiones en las siglas para obtener patrones como *AFL-CIO* y *3-D*. No tienen en cuenta las siglas que terminan en letras minúsculas excepto la *s*, y sólo detectan siglas con un dígito.
- *Algoritmo simple*. Usa un enfoque minimalista; excluye siglas con dígitos, puntos y espacios. Busca unidades que comiencen por una letra mayúscula, seguida de cero a ocho letras mayúsculas minúsculas, barras, guiones y que finalicen con una letra mayúscula.

2) Detección de formas desarrolladas

Los elementos que tiene en cuenta cada algoritmo para la detección de FD son:

- *Algoritmo contextual*. Encuentra las FD buscando coincidencias desde el último caracter de una sigla hacia atrás. Siempre guarda la ventana con las últimas 20 palabras que preceden la sigla, de manera que, cuando identifica un candidato a sigla, trata de encontrar la FD dentro de esta. En caso de no encontrarla, continuará la búsqueda en el texto que hay después de la sigla. No requiere de formas

canónicas, por tanto, trata con éxito textos como: ... *"is three dimensional. In 3D images..."*

Las reglas de FD remiten a una lista de 35 de palabras vacías como *and*, *for*, *of* y *the*, las cuales suelen omitirse en la formación de las siglas; *e.g.*: *CIIR* (*Center for Intelligent Information Retrieval*). El algoritmo trata de hallar una secuencia de palabras tal que parte de los primeros cuatro caracteres de cada palabra, que no sean una palabra vacía, concuerden con los caracteres de la sigla; *e.g.*: *Bureau of Personnel* (*BUPERS*). Además:

- Un caracter inicial de una palabra vacía puede coincidir con un caracter interno de una sigla; *e.g.*: *Department of Defense* (*DOD*).
- Una palabra vacía puede omitirse; *e.g.*: *Research Experience for Undergraduates* (*REU*).
- El primer, cuarto, quinto o sexto carácter de las palabras del candidato a FD puede concordar con los caracteres de la sigla; *e.g.*: *PostScript* (*PS*).

Las siglas que poseen dígitos reciben un tratamiento especial. El algoritmo intenta reemplazar el dígito y el caracter anterior o posterior con *n* repeticiones del carácter; *e.g.*: *MMM* por *3M*. Si no puede encontrar una FD para esta sigla transformada, entonces trata de encontrar la concordancia entre el dígito y el número escrito en letras, por ejemplo: *three dimensional* por *3D*. Los puntos en las siglas se ignoran cuando se buscan las FD.

Uno de los principales problemas del algoritmo contextual es su tendencia a tratar de hacer coincidir más de un caracter inicial de las palabras de la FD. Esto lleva al algoritmo a expandir *NIST* como *National Institute of Standards*, tomando la *t* de *Standards*, en lugar de *National Institute of Standards and Technology*. Otro problema, en particular con las siglas de dos letras, es su tendencia a hallar secuencias de palabras en minúscula con una concordancia falsa para la sigla; *e.g.*: *story from* para *SF*.

- *Algoritmo canónico-contextual*. Es una modificación del algoritmo contextual para tratar los dos problemas antes mencionados. En primer lugar, incluye reglas canónicas para restringir la aceptación de palabras minúsculas en la FD. Solamente se permite una forma desarrollada en minúsculas si un par sigla-FD cumple con una de las formas que se describen en la tabla 1. Una FD hallada por medio de reglas contextuales debe estar en mayúscula a excepción de las *stopwords*. En segundo lugar, el algoritmo trata de buscar, con criterio conservador, las concordancias entre múltiples caracteres en una FD, solucionando de esta forma el problema ilustrado anteriormente con *NIST*. Además, los guiones y las barras se tienen en cuenta dentro de las siglas, pero se pasan por alto al expandirlas. Si una FD está separada por guiones; *e.g.*: *Real-Time*, que hace parte de la sigla CRICCS *(Center for Real-Time and Intelligent Complex Computing Systems)*, el algoritmo puede tratar *Real Time* como dos palabras o como una sola palabra, sin necesidad de que exista una *T* en la sigla.

- *Algoritmo canónico.* Es un derivado del algoritmo canónico-contextual del cual sólo toma los pares sigla-FD que se encuentren en la forma canónica.

- *Algoritmo simple.* Busca suprimir gran parte de la complejidad del algoritmo contextual y sus derivados. Al igual que el algoritmo canónico, el algoritmo simple requiere que la sigla se encuentre en ciertos contextos, aunque acepta menos patrones canónicos para los pares sigla-FD y menos patrones de sigla. El algoritmo busca las formas presentadas en la *forma desarrollada canónica* de la tabla 1, en el orden en que se listan.

  Al momento de revisar la validez de un candidato a FD, el algoritmo tiene varios esquemas de concordancia sigla-FD. Cada uno de los cuales revisa repetitivamente las FD más cortas primero. Los esquemas de concordancia se llevan a cabo de la siguiente manera:

- *Mayúscula estricta*. Cada letra en la sigla debe estar representada, en orden, por una letra mayúscula en la FD. Esta debe comenzar con la primera letra de la sigla.
- *Minúscula estricta*. Cada letra en la sigla debe estar representada, en orden, por la primera letra de una palabra en la FD. Ésta debe comenzar con la primera letra de la sigla y no debe contener letras mayúsculas.
- *Mayúscula flexible*. La primera palabra debe comenzar con la primera letra de la sigla y la última palabra debe comenzar con una letra de la sigla. Este esquema es sumamente flexible, pudiendo llevar a formas desarrolladas donde algunas letras de la sigla no coincidan en absoluto.

SIGLAS	Algoritmos		
	Contextual	*Canónico-contextual / Canónico*	*Simple*
Patrones para las siglas	(U {sep})$^{+}$ *e.g.*: U.S.A U^{+} *e.g.*: USA D*U[DU]* *e.g.*: 3D,62A2A UUU^{+}L^{+} *e.g.*: JARtool UU^{+}L^{+}U^{+} *e.g.*: AChemS	(U {sep})$^{2-9}$S *e.g.*: U.S.A, U.S.A. 's U$^{2-9}$S *e.g.*: USA, USA's U*{dig}U^{+} *e.g.*: 3D, 3-D, I3R U^{+}L^{+}U^{+} *e.g.*: DoD U^{+}[/-]U^{+} *e.g.*: AFL-CIO	U[UL/-]$^{0-8}$U *e.g.*: USA, DoD, AFL-CIO
Mayúsculas *vs* minúsculas	Los dos primeros caracteres deben ser U, luego cualquier número de L en alguna parte, pero adyacente	L interna, o *s* final o *'s* DOD, DoD, DOD's	Debe comenzar y finalizar con U Puede tener L en otro sitio DOD, DoD
Dígitos	Cualquier número de dígitos en cualquier lugar	Sólo un dígito, en cualquier posición que no sea la final, *e.g.*: 3M, 2ATAF	Ninguno
Espacios y puntos	Después de letras mayúsculas	"." o ". + espacio" debe estar después de cada caracter, *e.g.*: N.A.S.A, N. A. S. A.	Ninguno
Barra o guión (/ ó -)	Ninguno. Tratados como espacio en la *tokenización*	Una barra o guión interior, *e.g.*: OB/GYN, CD-ROM	Cualquier número de barras o guiones en el interior, *e.g.*: OB/GYN, CD-ROM
Longitud máxima	No está explícita	9 caracteres alfanuméricos, más cualquier puntuación incluida o *s* final	10 caracteres incluyendo cualquier puntuación
FORMAS DESARROLLADAS			
Palabras vacías o *Stopwords* (*and, for, of, the*)	Lista fija de 35 *stopwords*	Lista fija de 40 *stopwords*	Ninguna
Palabras omitidas	Sólo *stopwords*	*Stopwords* o palabras que se encuentran después de guiones	Sólo las primeras y últimas palabras tienen que coincidir con caracteres en la sigla
Caracteres de *stopwords*	Como máximo uno, únicamente caracteres internos en la sigla		No aplicable
Prefijos	Si, asume que cualquier inicial hasta la quinta posición puede ser un prefijo		No aplicable
Caracteres procedentes de palabras que no son *stopwords*	Hasta 4 caracteres. Algoritmo *"greedy"*, prefiere tomar más	Hasta 4 caracteres. Algoritmo "conservador", prefiere tomar menos	Prefiere hasta 1 caracter. Puede tomar más si la palabra comienza por mayúscula
Forma desarrollada canónica	No aplicable	(se buscan en desorden) SIGLA (Forma desarrollada), Forma desarrollada (SIGLA) (Forma desarrollada) SIGLA, (SIGLA) Forma desarrollada SIGLA or Forma desarrollada, Forma desarrollada or SIGLA, SIGLA stands for Forma desarrollada SIGLA {is} an acronym for Forma desarrollada known as the SIGLA Forma desarrollada "SIGLA", "SIGLA" Forma desarrollada	(se buscan en orden) Forma desarrollada (SIGLA) Forma desarrollada or SIGLA Forma desarrollada, or SIGLA Forma desarrollada, SIGLA SIGLA (Forma desarrollada) SIGLA, Forma desarrollada
Uso de mayúsculas o minúsculas	La forma desarrollada puede aparecer en su totalidad en letras minúsculas	Canónico: todas pueden ser minúsculas Contextual: sólo las *stopwords* pueden ser minúsculas, el resto deben ser mayúsculas	Se permiten las minúsculas, pero con reglas más estrictas que las de las mayúsculas; la letra inicial de cada palabra de la forma desarrollada debe coincidir con la letra que ocupa idéntica posición en la sigla.
Números	En letras o en dígitos		Sin números

Tabla 1. Características de las siglas y sus FD según el tipo de algoritmo[7] (Larkey et al., 2000)

[7] El *superescript* + indica una o más ocurrencias de un símbolo; * indica 0 ó más ocurrencias; los *superescripts* numerados indican un número específico o rango de ocurrencias; U significa una letra mayúscula; L significa una letra minúscula; D significa un dígito; S significa un final opcional *s* o *'s;*{sep} es un punto o punto seguido por un espacio, y {dig} es un número entre 1 y 9, opcionalmente seguido de un guión. Los términos en corchetes son alternativos

c. Evaluación

Para la evaluación de los algoritmos, Larkey *et al.* tomaron un corpus de 936.550 páginas web de instituciones militares y gubernamentales de los Estados Unidos, el cual procesaron para incluir en la base de datos *Acrophile*. De este conjunto se escogieron al azar 170 páginas para buscar los pares de siglas-FD. Como resultado se obtuvieron 353 pares de siglas-FD, de los cuales 10 tenían símbolos como & y /. Ninguna de las siglas presentaba números o guiones. Las variaciones en las FD consideradas como correctas fueron la omisión o adición de la 's' y las diferencias en la puntuación.

Los siguientes son los valores de precisión y exhaustividad para los 4 algoritmos en las 353 siglas del test (328 de las cuales tienen una longitud igual o mayor a 3 caracteres).

Algoritmo	**Todas las siglas**		**Siglas de longitud > 2 caracteres**	
	Precisión	Exhaustividad	Precisión	Exhaustividad
Contextual	.89	.61	.96	.60
Canónico-contextual	**.87**	**.84**	.92	.84
Canónico	.96	.57	.99	.59
Simple	.94	.56	.99	.57

Tabla 2. Precisión y exhaustividad en el corpus de evaluación (Larkey et al., 2000)

Los cuatro algoritmos fallaron en la detección de 16 casos debido a que la FD estaba a una distancia superior a 20 palabras; es decir, demasiado lejos de su sigla correspondiente. Sin embargo, los autores dejan claro que no estaba dentro de sus expectativas que alguno de sus algoritmos lo consiguiera.

El algoritmo de mejor rendimiento es el *canónico-contextual*. No obstante, para los autores, sus resultados no son comparables con la precisión y la exhaustividad alcanzadas por los sistemas de Taghva & Gilbreth y de Yeates, puesto que éstos empelan corpus y criterios de exactitud diferentes.

4. Acromed

Putstejovsky *et al.* (2001) desarrollaron un sistema llamado *Acromed*, una de las herramientas diseñadas para procesar y extraer información de los *abstracts* de la base

de datos *Medline*. Estos autores afirman que este sistema se diferencia de los sistemas de extracción de siglas preexistentes (Taghva & Gilbreth, 1999; Yeates, 1999 y Larkey, 2000) en que el algoritmo de reconocimiento de siglas incluye un análisis sintáctico superficial o *shallow parsing* de los textos.

a. Patrones

Los pares de candidatos deben coincidir con el patrón "FD (sigla)". El caracter inicial de la primera palabra de la FD debe coincidir con el caracter inicial de la sigla.

b. Técnica de extracción de siglas

La estrategia de extracción de siglas de *Acromed* tiene dos vías. La primera considera el problema del reconocimiento de pares FD-sigla como el problema de encontrar dos cadenas de caracteres en un texto que coincidan con ciertas expresiones regulares, a lo que se denomina *algoritmo de expresión regular*.

Los autores han desarrollado un patrón muy restringido para el par FD-sigla:

$String_i$ ($String_j$)

Donde
"#" significa el límite de una oración
$String_i$ representa la FD
($String_j$) representa la sigla

Debido a que el patrón usado era demasiado limitado, Pustejovsky *et al.*, decidieron incluir la posibilidad de buscar las FD en la ventana o contexto derecho además del izquierdo, como medida para mejorar la exhaustividad.

La segunda vía consiste en el refinamiento del paso anterior. Aunque el problema básico es el mismo (es decir, dos cadenas de caracteres se comparan para decidir si una es la FD de la otra), la extensión y límites del contexto donde se busca la FD son totalmente diferentes.

Acromed usa dos mecanismos para la extracción de información como son el preproceso de textos y la anotación sintáctica. Posteriormente, busca una sigla objetivo en un contexto tal y como se ilustra a continuación:

EXP_i, EXP_j, T_ACRONYM, EXP_k, EXP_m, donde las expresiones son cadenas de caracteres etiquetadas y T_ACRONYM es otra expresión (generalmente una cadena etiquetada) como en:

1. [[‘the’, ‘DT’], [‘performance’, ‘NN’], [‘of’, ‘IN’], [‘an’, ‘DT’], [‘automatic’, ‘JJ’], [‘speech’, ‘NN’], [‘recognition’, ‘NN’], [‘NX’],
2. [‘(‘,’(‘],
3. [[‘ASR’, ‘NN’], ‘NNX’]
4. [‘)’, ‘)’]

Con el ejemplo anterior Pustejovsky *et al*. muestran 4 expresiones que son la entrada para el detector de pares FD-sigla. Bajo este modelo, las cadenas de caracteres: *“The performance of an automatic speech recognition*, *ASR”* serán usadas como entrada para la expresión regular para el reconocimiento del par FD-sigla.

Según Pustejovsky, este diseño permite restringir considerablemente el contexto de búsqueda de la FD. En un algoritmo que considere únicamente las cadenas de caracteres y su contexto, se debe establecer una ventana o límite arbitrario. Con el análisis sintáctico superficial, el límite se establece naturalmente gracias a las propiedades del lenguaje. Con esta estrategia, se especifica que la FD es un sintagma nominal que está cerca del candidato a sigla (o sigla “objetivo”). Dentro del contexto de un candidato a sigla se pueden establecer restricciones como signos de puntuación y coordinación de sintagmas nominales.

Estos autores utilizan un autómata de estado finito que usa las expresiones, verificando sus tipos; es decir, sintagma nominal, sintagma verbal o signo de puntuación. Si se encuentra un candidato par FD-sigla, entonces las cadenas de caracteres

correspondientes a ambas expresiones se suministran a la cadena de búsqueda de siglas (la estrategia previa), la cual decidirá si una subcadena de la FD concuerda con la sigla. Si es así, se considerará como una identificación positiva y se almacenará en la BD de siglas.

Los primeros experimentos con el mecanismo de *restricciones sintácticas* usaron sólo el siguiente patrón:

1. T_LF_Noun Phrase$_1$ (T_A_Noun Phrase$_2$)

donde T_LF significa objetivo donde hallar la FD y T_A significa objetivo o candidato a sigla.[8]

Los experimentos posteriores usaron *restricciones sintácticas modificadas* y adicionaron los siguientes patrones:

1. T_A_Noun Phrase$_1$ (T_LF_ Phrase$_2$)
2. T_A_Noun Phrase$_1$, T_LF_Noun Phrase$_2$
3. (T_A_Noun Phrase$_1$) T_LF_Noun Phrase$_2$

c. Evaluación

Para la fase de evaluación del sistema se empleó un corpus de entrenamiento y otro de evaluación.[9] El primero contenía 86 *abstracts* de la base de datos de *Medline* y 155 pares sigla-FD, mientras que el segundo contenía 100 *abstracts* y 173 pares sigla-FD.

En el corpus de entrenamiento, *Acromed* recuperó 123 pares FD-sigla, de los cuales 106 eran correctos. Los resultados de precisión y exhaustividad son similares a los obtenidos en otras investigaciones sobre el tema.

[8] LF *(long form)* es equivalente a FD (forma desarrollada de la sigla). T_A equivale a *"target acronym"* (sigla objetivo o candidato a sigla).

[9] Estos autores toman como punto de referencia para la evaluación de su sistema el *Gold Standard*, formado por 149 pares de sigla-FD.

En el corpus de evaluación, *Acromed* empleó los algoritmos de expresión regular y restricción sintáctica. Con el primer algoritmo recuperó 117 pares de sigla-FD, de los cuales 106 eran correctos. Con el segundo algoritmo, recuperó 105 pares, de los que 104 eran correctos.

Algunos errores en la detección de pares FD-sigla se asocian al problema de la delimitación de la ventana para la búsqueda de la FD; *e.g.*: *p16= products*, extraído de: *"which encodes two gene products (p16(INK4a) and p19(ARF))"*.

Los resultados obtenidos por *Acromed* con respecto a la precisión y la exhaustividad fueron los siguientes:

	Corpus de entrenamiento		**Corpus de evaluación**	
	Precisión	Exhaustividad	Precisión	Exhaustividad
Expresión regular	88.1%	73.2%	90%	63%
Restricciones sintácticas	97.2%	72.5%	99%	61.9%
Restricciones sintácticas modificadas	94.6%	82.5%	**98.3%**	**72%**

Tabla 3. Precisión y exhaustividad de Acromed (Pustejovsky et al., 2001)

5. Sistema para gestión de variación terminológica

Nenadić *et al.* (2002) consideran las siglas como un fenómeno de variación terminológica muy común. Las siglas pueden ser variantes terminológicas de tipo léxico-semántico, puesto que se usan como sinónimos de sus FD correspondientes; o pueden ser variantes terminológicas de tipo pragmático, ya que facilitan la lectura de los textos científicos.

Estos autores consideran que expertos como los biólogos moleculares crean frecuentemente siglas específicas que usan localmente (dentro de un artículo científico) o dentro de todo su campo de especialidad.

Las siglas, al igual que los términos, adolecen de los siguientes problemas:

- *variación*. Un mismo término puede tener varias siglas; *e.g.*: *NF kappa*, *NF kB*);
- *ambigüedad.* Una misma sigla puede referir a conceptos diferentes; *e.g.*: *GR* puede referir tanto a *glucocorticoid receptor* como a *glutathione reductase*.

Nenadić *et al.* sólo tratan el problema de variación de las siglas. Consideran que el problema de la ambigüedad puede resolverse simplemente con el uso de la última FD introducida en el texto, en caso de que haya una. Si no hay una FD introducida, entonces deben usarse los métodos generales para la desambiguación de términos.

a. Técnica de extracción de siglas

Para localizar los candidatos a FD, el sistema de Nenadić *et al.* se basa en los patrones sintácticos que se usan principalmente en los artículos científicos. Una vez se detecta que una secuencia de palabras coincide con un patrón *x*, se recupera y se analiza morfológicamente con el propósito de descubrir la relación entre la sigla y su FD.

El método de adquisición de siglas consta de tres pasos, a saber:

1) Recuperación de las FD

En este paso se explora el texto para la búsqueda de candidatos a FD. Varios patrones de FD se han identificado manualmente para describir varios contextos de introducción de una sigla, a saber:

- FD a la izquierda; *e.g.: 9-cis retinoic acid (9cRA)*
- FD a la derecha; *e.g.: MIBP (Myc-intron-binding peptide).*

Para estos autores más del 90% de las ocurrencias corresponden al patrón "FD a la izquierda"; suelen introducirse mediante el uso de paréntesis; *e.g.*: *tumor necrosis factor alpha (TNF-alpha)* y, raras veces, mediante el uso de un formato similar a la aposición; *e.g.*: *...enzyme-linked immunosorbent assay, ELISA, ...*).

2) Concordancia entre siglas y FD

En este paso se aplica un conjunto de patrones de formación de siglas para lograr la concordancia entre un candidato a FD y su sigla. En general, las siglas se forman mediante la selección de caracteres iniciales de las palabras de la FD. Sin embargo, se ha observado que en el campo de la Biología molecular las iniciales de las formas de combinación también se usan con el mismo propósito. Las formas de combinación son afijos específicos (principalmente prefijos e infijos como: *acetyl*, *trans*, *di*, e *hydro*), que se usan con regularidad en los patrones de formación de términos; *e.g.*: *chloramphenicol acetyltransferase (CAT)*. En el momento de buscar la concordancia entre la FD y la sigla se usa un diccionario de formas de combinación del ámbito de la Biología molecular.

El método básico de concordancia entre siglas y sus FD mejora si se tienen en cuenta los siguientes fenómenos relacionados con las FD:

- *Inserción*. Una palabra está presente en la FD, pero no ha sido usada en la formación de la sigla; *e.g.*: *thyroid hormone receptor (TR)*
- *Omisión*. Una palabra falta en la FD, aunque se usa al momento de formar la sigla; *e.g.*: *[human] estrogen receptor (hER)*
- *Sigla en plural*. Se establece un plural para una sigla; *e.g.: retinoid x receptors (RXRs)*
- *Sigla recursiva*. La FD de una sigla contiene a su vez otra sigla o abreviatura; *e.g.*: *CREB-binding protein (CCBP)*
- *Siglas coordinadas*. Las siglas se definen dentro de una estructura coordinada*; e.g.: estrogen (ER) and progesterone (PR) receptors*
- *Sigla parcial*. Una sigla contiene una parte de su FD, normalmente palabras griegas o latinas; *e.g.*: *retinoid x receptor alpha (RXR alpha)*
- *Variación estructural*. Se define una sigla y posteriormente se realiza una transformación morfológica/estructural en su FD; *e.g.*: *day of hatching (HD)*
- *Siglas fórmula*. Una sigla contiene una parte de una fórmula química; *e.g.: 1alpha,25-dihydroxyvitamin D3 [1,25 (OH) 2D3].*

Los fenómenos anteriormente listados se consideran cuando el método básico de concordancia (es decir, la concordancia de caracteres de la sigla con los constituyentes de un candidato a FD) no arroja un resultado positivo.

Finalmente, el paso anterior produce una lista de siglas que concuerdan con sus FD.

3) Agrupamiento de siglas

Por último, en el tercer paso, los autores tratan de establecer las clases de variantes de una sigla. En primer lugar, tanto las siglas como sus FD se normalizan con respecto a sus rasgos ortográficos, morfológicos, sintácticos y léxico-semánticos. En particular, las siglas en plural como *NRs (nuclear receptors)* se hacen concordar con la correspondiente sigla en singular *NR (nuclear receptor)*. Todas las siglas que comparten una FD normalizada conforman un clúster o agrupación de siglas.

b. Evaluación

La evaluación se llevó a cabo a partir de dos corpus creados a partir de la BD *Medline*, conformados por 2.008 y 6.323 *abstracts*, respectivamente.

Para la evaluación sobre la adquisición de siglas se tomó una muestra aleatoria de 50 *abstracts* del primer corpus. La siguiente tabla muestra algunos ejemplos de siglas reconocidas automáticamente.

Sigla(s)	FD
RAR alpha	Retinoic acid receptor alpha
RAR-alpha	
RARA	
RARa	
RARs	Retinoic acid receptors Retinoic acid receptor
RAR	
RT-PCR	Reverse transcription PCR
TR	Thyroid hormone receptor Thyroid hormone receptors Thyroid receptor
TRs	
9-c-RA	9-cis-retinoic acid 9-cis retinoic acid
9cRA	
ES	Ewing sarcoma Ewing's sarcoma Ewings sarcoma

Tabla 4. Ejemplos de siglas reconocidas por el sistema de Nenadić et al. (2002)

La precisión de este método es muy alta, ubicándose en un rango entre 94% y 99%, dependiendo del tamaño del corpus. Aunque la exhaustividad del 73% no es un resultado despreciable, Nenadić sostiene que podría mejorarse, dado que se han identificado patrones adicionales durante la fase de evaluación manual.

En la siguiente tabla se presentan los resultados de la evaluación en los diferentes corpus.

Corpus / **Siglas**	**2.008 *abstracts***	**6.323 *abstracts***	**50 *abstracts***
Número de siglas diferentes reconocidas	1.015	2.343	66
Número de siglas reconocidas correctamente	992	2.314	62
Número de siglas introducidas	-	-	85
Precisión	97.73%	98.76%	93.94%
Exhaustividad	-	-	72.94%

Tabla 5. Evaluación en los diferentes corpus con el sistema de Nenadić et al.

6. A simple algorithm

Schwartz & Hearst (2003) implementaron un algoritmo simple para extraer pares de sigla-FD presentes en textos biomédicos.

El sistema ejecuta dos tareas: la primera consiste en la extracción de los pares de candidatos sigla-FD, mientras que la segunda consiste en la identificación de la FD correcta a partir de los candidatos presentes en el contexto de la sigla.

a. Patrones

Los patrones para seleccionar un candidato a sigla son:

- 2 a 10 caracteres
- máximo 2 palabras
- mínimo una letra
- primer caracter alfanumérico.

Los patrones para seleccionar un candidato a FD son:

- Una FD debe aparecer inmediatamente antes o después de su sigla correspondiente, es decir, en la misma oración y no debe tener más de (|A|+5), (|A|*2) palabras. Donde |A| es el número de caracteres de la sigla.

El método de selección de los candidatos a sigla, al igual que en muchos de los métodos existentes, se determina por su adyacencia a un paréntesis; es decir:

- FD (sigla)
- Sigla (FD)

Según estos autores, en la práctica la mayoría de los pares de candidatos se ajustan al patrón FD (sigla), de ahí que sea el patrón que emplean en su estudio. Además, subrayan que los candidatos a FD contiguos a la sigla son los únicos que se tienen en cuenta.

b. Técnica de extracción de siglas

Cuando se detecta un candidato a sigla, el algoritmo busca su FD en el contexto que se encuentra a derecha e izquierda. El algoritmo trata de encontrar la concordancia entre cada caracter de la sigla y de la FD moviéndose a la izquierda, comenzando desde el final de ambas cadenas de caracteres. El algoritmo acierta si el primer caracter de la sigla coincide con el primer caracter de cada palabra de la FD.

c. Evaluación

El algoritmo considera dos tipos de patrones, a saber: "FD (sigla)" y "sigla (FD)". La FD debe estar contigua a la sigla. El algoritmo se evaluó en dos corpus diferentes. Por un lado, se probó con una versión corregida del "*Gold Standard*" de Pustejovsky *et al.*, que contiene 168 pares de sigla-FD. En este caso el sistema identificó 143 pares de sigla-FD, de los cuales 137 eran correctos. Los 31 pares sigla-FD restantes no fueron identificados, pues no presentaban una coincidencia exacta entre sus caracteres; *e.g.*:

CNS1 (cyclophilin seven suppressor); *ATN (anterior thalamus)*. Este resultado se traduce en 96% de precisión y 82% de exhaustividad.[10]

Por otro lado, el algoritmo se evaluó en un corpus de 1.000 *abstracts* extraídos de *Medline*, los cuales contenían 954 pares sigla-FD. El rendimiento fue de 95% de precisión y 82% de exhaustividad.

A modo de síntesis, se presenta la siguiente tabla comparativa del rendimiento de los sistemas de detección de siglas basados en patrones.

Sistema	Autor	Año	Corpus	Área	Precisión	Exhausti-vidad
AFP	Taghva & Gilbreth	1999	17 documentos técnicos (463 siglas-FD)	Medio ambiente	98%	93%
TLA	Yeates	1999	10 documentos técnicos	Informática	68%	91%
Acrophile	Larkey *et al.*	2000	170 páginas web (353 siglas-FD)	Militar-gubernamental	87%	84%
Acromed	Pustejovsky *et al.*	2001	100 *abstracts* de la BD *Medline* (173 siglas-FD)	Biomedicina	98%	72%
Sistema de gestión de variación terminológica	Nenadić *et al.*	2002	50 *abstracts* de *Medline* (85 siglas-FD)	Biomedicina	94%	73%
A Simple Algorithm	Schwartz & Hearst	2003	100 *abstracts* de *Medline* (168 siglas-FD)	Biomedicina	96%	82%
			1.000 *abstracts Medline* (954 siglas-FD)		95%	82%

Tabla 6. Rendimiento de los sistemas de detección de siglas basados en patrones

Aunque los resultados logrados por el sistema de Taghva & Gilbreth son superiores, la mayoría de los autores coinciden en afirmar que dichos resultados no son comparables dado que su algoritmo no tuvo en cuenta las siglas de dos caracteres y se evaluó en un corpus muy pequeño, de tan solo de 17 textos. Puede decirse entonces que los dos métodos con mejor rendimiento dentro del grupo de sistemas basados en patrones son los de Schwartz & Hearst y Larkey *et al.*, respectivamente.

[10] Schwartz & Hearst manifiestan que los resultados de su algoritmo son bastante parecidos a los alcanzados por otros sistemas más complejos como los de Pustejovsky (72% de exhaustividad y 98% de precisión) o Chang (83% de exhaustividad y 80% de precisión).

B. Métodos basados en estadística y aprendizaje máquina

Las investigaciones más recientes sobre técnicas de extracción de siglas apuntan al desarrollo de métodos estadísticos y de aprendizaje máquina.

Los métodos estadísticos se basan en la frecuencia de las siglas en un corpus. Dentro de esta clase destacan los trabajos de Chang *et al.* (2002) y Adar (2002).

Los algoritmos de aprendizaje usan ejemplos, atributos y valores para reconocer y clasificar siglas. Tienen la capacidad de mejorar con la experiencia (entrenamiento). Entre los autores que han desarrollado sistemas basados en este método destacan Young (2004), Zahariev (2004), Dannélls (2005) y Nadeau & Turney (2005).

Además de los dos métodos anteriores, existen los métodos híbridos, es decir, aquellos que combinan la estadística con el aprendizaje máquina. Dentro de estos métodos destaca el trabajo de Park & Byrd (2001).

1. Métodos basados en técnicas estadísticas

1.1. Diccionario de abreviaciones en línea

Chang *et al*. (2002) desarrollaron un método de detección de siglas basado en un algoritmo de regresión logística, que usa un conjunto de rasgos para describir los diferentes patrones presentes en las siglas.

El conjunto de rasgos empleados por Chang *et al*. para la descripción de las siglas es el siguiente:

Rasgo	Descripción
Minúscula *vs.* mayúscula	% de letras en la sigla en minúscula
Comienzo de palabra	% de letras alineadas al comienzo de una palabra
Final de palabra	% de letras alineadas al final de una palabra
Límite de sílaba	% de letras ubicadas en un límite de sílaba
Después de letra alineada	% de letras alineadas inmediatamente después de otra letra
Letras alineadas	% de letras que están alineadas en la sigla
Palabras omitidas	Número de palabras en la FD no alineadas con la sigla
Letras alineadas por palabra	Número promedio de letras alineadas por palabra
CONSTANTE	Normalización constante por algoritmo de regresión logística

Tabla 7. Vector de rasgos de una sigla(Chang et al., 2002)

a. Heurística

- el algoritmo solo considera los candidatos que aparezcan entre paréntesis de acuerdo con el patrón "FD (sigla)"
- dentro del paréntesis se recuperan las palabras que se encuentren antes de una coma o un punto y coma
- una sigla debe contener una letra como mínimo
- el contexto o ventana de búsqueda de la FD es de 2*|A| palabras.

b. Técnica de extracción de siglas

Para Chang *et al.* el proceso de detección de las siglas consta de cuatro pasos, a saber:

1) búsqueda de candidatos a sigla

Para la búsqueda de los candidatos a sigla se emplea el patrón "FD (sigla)". Dentro de los paréntesis sólo se recuperan las cadenas de caracteres que están antes de una coma o un punto y coma. Se rechazan los candidatos a sigla que no posean ninguna letra. Para cada candidato se salvan las palabras que se encuentran antes del paréntesis (prefijo), de manera que se pueda buscar en ellas la FD de la sigla.

2) alineación de los candidatos a sigla con el texto que los precede (ventana)

Las letras del candidato a sigla se alinean con las del prefijo o texto que lo precede. Este paso es equivalente al procedimiento que ejecuta el algoritmo *Longest common subsequence (LCS)*, empleado en investigaciones como la de Taghva & Gilbreth.

3) conversión de las alineaciones en un vector de rasgos

Se calculan los vectores de rasgos que describen cuantitativamente al candidato par sigla-FD. Para llevar a cabo esta tarea, Chang *et al.* han usado las 9 características que más información aportan sobre el candidato. Estas características se establecieron a partir de las observaciones realizadas sobre un corpus de *abstracts* de *Medline*.

4) puntuación de las alineaciones mediante un algoritmo de aprendizaje máquina

Se utiliza la puntuación de las alineaciones mediante un algoritmo de aprendizaje máquina. Para el entrenamiento de este algoritmo se empleó un corpus de 1.000 candidatos a sigla seleccionados aleatoriamente de *abstracts* de *Medline*. A partir de estos *abstracts* se identificaron 93 siglas y se anotó manualmente la alineación entre las siglas y sus prefijos. Luego se generaron todas las alineaciones posibles en el corpus de los 1.000 candidatos, lo que llevó a la creación del corpus de experimentación. Los tipos de alineaciones que se encontraron fueron:

- alineación de siglas incorrectas
- alineación correcta de siglas correctas
- alineación incorrecta de siglas correctas.

Todas estas alineaciones se convirtieron en vectores de rasgos que sirvieron para entrenar el clasificador de regresión logística. Como resultado se obtuvo una lista de candidatos a sigla junto con sus FD y puntajes. El sistema considera correcto un par sigla-FD cuando coincide exactamente con el *Gold Standard* de *Medstract*. Adicionalmente, el sistema sólo toma el puntaje más alto para cada sigla.

Chang *et al.* almacenaron en una BD relacional aquellas siglas con un puntaje mayor o igual a 0.001. La BD se encuentra en un servidor web;[11] permite búsquedas por sigla o por palabra además de buscar siglas en textos directamente suministrados por el usuario.

[11] *Cf.* http://abbreviation.stanford.edu/

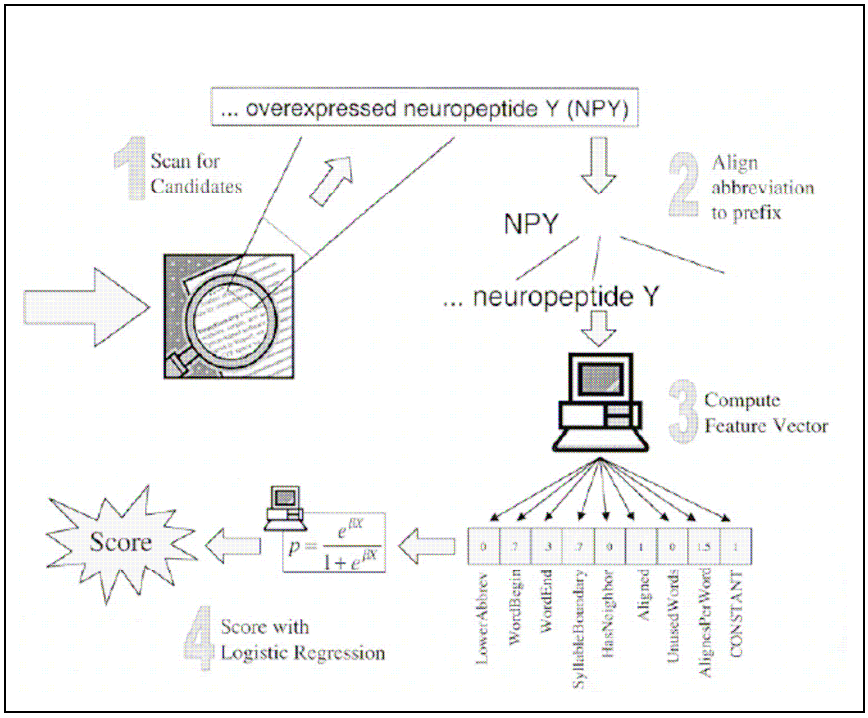

Fig. 3. Arquitectura del sistema (Chang et al., 2002)

c. Evaluación

Chang *et al*. evaluaron su algoritmo contra el *Gold Standard* de *Medstract*, el cual contiene 168 siglas extraídas de *Medline* y anotadas por expertos. De éstas el sistema identificó correctamente 140, alcanzando 80% de precisión y 83% de exhaustividad.

Los aportes de Chang *et al.* a la investigación en este campo son:

- el desarrollo de un nuevo algoritmo para la identificación de siglas
- la elaboración de un conjunto de rasgos descriptivos de las siglas, y
- la creación de un nuevo diccionario de abreviaciones en línea a partir de los *abstracts* de la BD *Medline*.

1.2. A Simple and Robust Abbreviation Dictionary (SaRAD)

SaRAD es el sistema desarrollado por Adar (2002) orientado a la creación de entradas agrupadas y a la generación de reglas de clasificación para la desambiguación de FD. El

sistema emplea una técnica modificable que limita el espacio de búsqueda y optimiza el proceso de creación del diccionario.

a. Heurística

- El patrón más común es "FD (sigla)"[12]
- Una sigla es una palabra o un conjunto de palabras separadas por guiones donde hay al menos una letra mayúscula
- Para determinar la ventana de la FD se seleccionan como máximo un número de palabras igual a *n* + palabras *buffer* antes del paréntesis; donde *n* es el número de caracteres de la sigla y el *buffer* es cuatro.[13]

b. Técnica de extracción de siglas

En general, puede decirse que la creación del diccionario *SaRAD* implica el uso de varios módulos. El primero de ellos se encarga de procesar el corpus para la extracción de las siglas y la generación de los candidatos a FD. El segundo toma todas las FD y las agrupa por sigla. Los resultados de este paso se usan para hacer remisiones entre siglas y para agrupar las FD.

Posteriormente, se efectúa un segundo agrupamiento de los documentos de *Medline* basado en el análisis de los *Medical Subject Headings* (MeSH) para desambiguar las siglas y completar el diccionario.

[12] Según el autor esta conclusión se desprende de un análisis de 5.000 documentos de la base de datos *Medline.*

[13] El *buffer* se usa para dar cuenta de las palabras que a veces no se incluyen dentro de la sigla. Esto es importante en casos como el de la sigla *AABB* donde la FD (*American Association of Blood Banks*) incluye la preposición *"of"*.

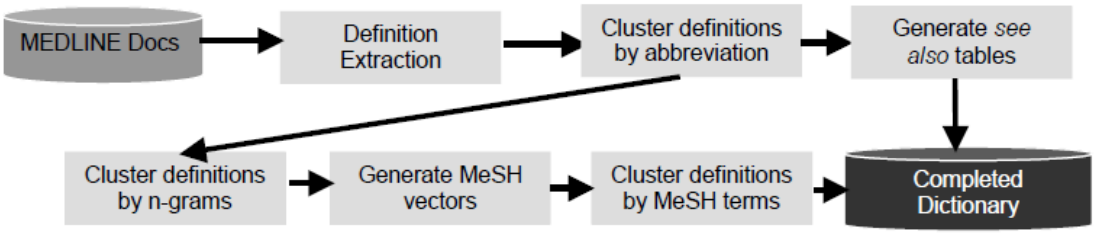

Fig. 4. Arquitectura general del sistema (Adar, 2002)

En particular, el proceso de extracción de siglas empleado en la construcción de *SaRAD* consta de los siguientes pasos:

- búsqueda de las siglas en el texto fuente
- búsqueda de las FD en la ventana de texto que rodea a la sigla
- generación de las rutas *(paths)* a través del texto que pueden definir la sigla
- puntuación de las rutas para determinar cuál es el mejor candidato a FD.

En cuanto al establecimiento de las ventanas para la búsqueda de las FD, existen muchas formas posibles de expandir una sigla en un texto, siendo la más común la que corresponde al siguiente patrón:

<texto> FD (sigla) <texto>

De acuerdo con las observaciones hechas en el corpus de *Medline*, se desarrolló el módulo de extensión para localizar una sigla dentro de un paréntesis y una ventana de texto antes de este, tal y como se representa en el siguiente ejemplo:

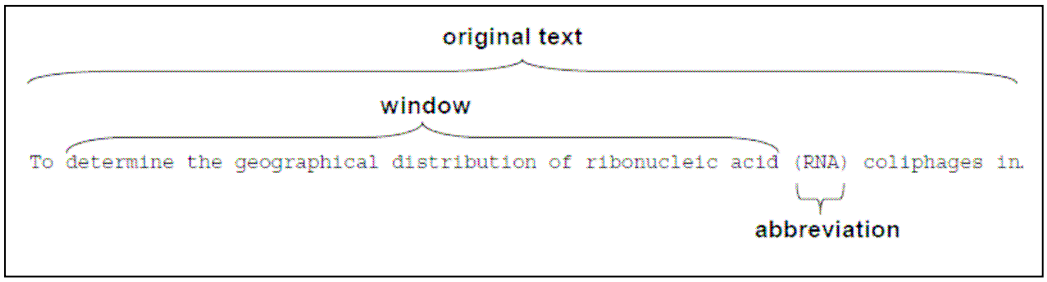

Fig. 5. Muestra de una ventana de texto para la búsqueda de la FD de RNA. (Adar, 2002)

Después de establecer la ventana de texto, se buscan los candidatos a FD, tarea que se ejecuta en la fase denominada "generación de las rutas o *paths*". Esto se logra mediante la búsqueda hacia adelante de los caracteres que coinciden "en orden" con los de la sigla. Este proceso permite crear la ruta que conduce a la ubicación de los caracteres de la sigla dentro del texto. La siguiente figura muestra tres rutas que corresponden a igual número de candidatos a FD de la sigla RNA dentro de la ventana *"determine the geographical distribution of ribonucleic acid"*.

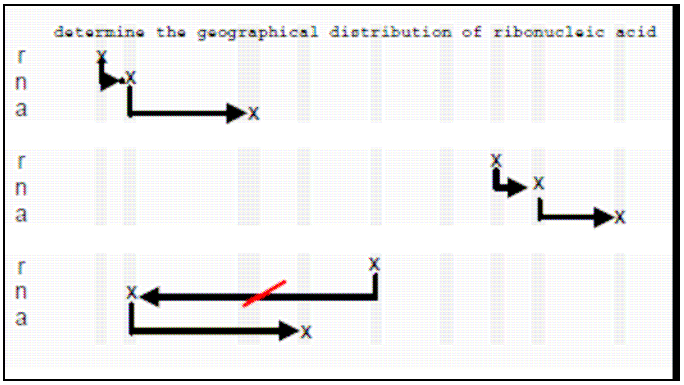

Fig. 6. Procedimiento del algoritmo para hallar la FD en la ventana de texto adyacente. (Adar, 2002)

Posteriormente, a las rutas encontradas se les asigna un puntaje para determinar si existe o no una FD adecuada. Dado que la aplicación de esta regla no es suficiente por sí sola para la identificación de las siglas, se requiere la aplicación de otras reglas como:

- por cada caracter de la sigla que se encuentre al comienzo de una palabra de la FD se agrega 1 punto
- por cada palabra extra entre la FD y el paréntesis donde se encontró la sigla se resta 1 punto

- a las FD que se encuentren justo al lado del paréntesis se agrega 1 punto extra
- El número de palabras de la FD debería ser menor o igual que el número de caracteres de la sigla. Por cada palabra extra se resta 1 punto.

El punto de equilibrio para las FD correctas es cero. Si el puntaje de un candidato a FD es mayor que cero, se considera que éste tiene una alta probabilidad de ser la FD correcta. La tabla 8 muestra un caso de puntuación para los candidatos a FD de la sigla RNA.

	Score
determine the geographical distribution of ribonucleic acid	
determine the geographical distribution of ribonucleic acid	-4
determine the geographical distribution of ribonucleic acid	-4
determine the geographical distribution of ribonucleic acid	-2
determine the geographical distribution of ribonucleic acid	-2
determine the geographical distribution of ribonucleic acid	-2
determine the geographical distribution of ribonucleic acid	0
determine the geographical distribution of ribonucleic acid	0
determine the geographical distribution of ribonucleic acid	1
determine the geographical distribution of ribonucleic acid	1
determine the geographical distribution of ribonucleic acid	3*

Tabla 8. Ejemplo de puntuación para la selección de la FD (Adar, 2002)

Una vez realizados los pasos que corresponden al primer módulo, cabe agrupar las FD relacionadas, que normalmente son las que están en plural; *i.e.*: *Estrogen Receptor* y *Estrogen Receptors*.

El sistema de Adar emplea dos técnicas de *clustering* diferentes. La primera, basada en n-gramas, se encarga de buscar definiciones con raíces similares. La segunda utiliza la lista de descriptores *Medical Subject Headings (MeSH)* para la creación del *cluster* de FD.

La técnica específica de n-grama que se utiliza es la de trigramas, la cual segmenta cada FD en grupos de tres letras que se comparan unas con otras, por ejemplo, la FD "ABCDE" contiene el grupo de trigramas (ABC, BCD, CDE).

La segunda técnica consiste en tomar cada *cluster*, encontrar los documentos iniciales de los que se extrajeron las definiciones y generar un vector que representa los términos de *MeSH*.

La parte final del análisis para crear el diccionario SaRAD consiste en crear las remisiones o referencias cruzadas entre las FD relacionadas. Esto es particularmente útil para unidades con variantes tipográficas como el uso de mayúsculas, por ejemplo: ACH, AcH, ACh, Ach, abreviaturas de *acetylcholine.*

Para efectuar las remisiones de las FD y generar la lista de "véase también" en el diccionario, se buscan todas las FD equivalentes en un rastreo a través de todo el diccionario.

Finalmente, para la desambiguación de las siglas se reutilizan los vectores de *MeSH* generados anteriormente para el agrupamiento de las FD.

c. Evaluación

El sistema es bastante limitado puesto que sólo tiene en cuenta los candidatos que se ajustan al patrón "FD (sigla)". Al aplicarse el algoritmo al *Gold Standard* de *Acromed*, el sistema halló 144 siglas. Teniendo en cuenta que el punto de equilibrio para la selección de las FD correctas es cero, el sistema alcanzó una precisión de 86% y una exhaustividad de 88%.

2. Métodos basados en algoritmos de aprendizaje máquina

2.1 Teoría universal de la formación de siglas

En la "Teoría universal de la formación de siglas", Zahariev (2004), las siglas se consideran un fenómeno universal cuya formación se rige por preferencias lingüísticas basadas en reglas a nivel de caracteres, fonemas, palabras y frases.

La teoría universal de la formación de siglas se desarrolla a partir de los ejemplos tomados de 15 lenguas con sistemas de escritura diferentes como son: inglés, español, francés, alemán, finlandés, italiano, húngaro, rumano, ruso, búlgaro, hebreo, árabe, farsi, chino y japonés.

El trabajo de Zahariev apunta a la solución de la adquisición y la desambiguación, los dos principales problemas en el tratamiento automático de las siglas. Para la solución de cada problema, se emplea un algoritmo de aprendizaje máquina.

En general, se propone un enfoque modular para el tratamiento de las siglas; para ello ejecutan las siguientes tareas:

- identificación de siglas
- identificación de FD
- concordancia de siglas-FD
- desambiguación de siglas.

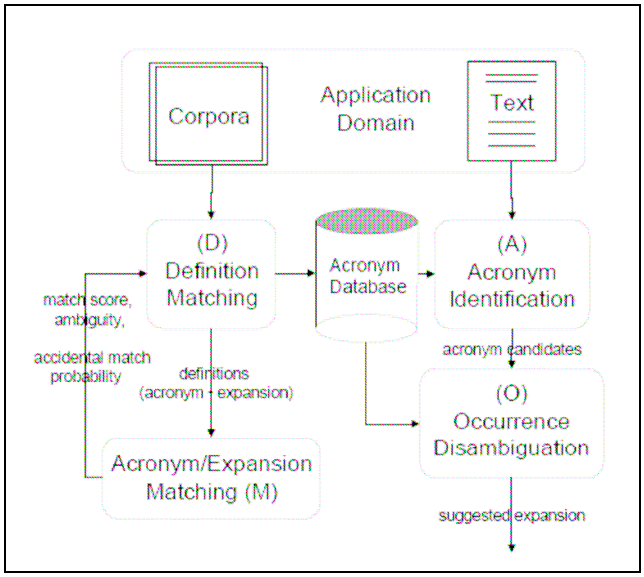

Fig. 7. Enfoque modular para la adquisición automática de siglas (Zahariev, 2004)

a. Patrones

El sistema de Zahariev reconoce candidatos a sigla que cumplen, en general, con algunas de las siguientes características:

- letras mayúsculas
- puntuación interior, *e.g.*: U.S.A, TCP/IP, S-MIME, MIB-s, etc.

Sin embargo, el sistema descarta aquellas siglas que contienen dígitos y caracteres diferentes a la barra y el guión; *e.g.*: 3M, MP+, etc.

A pesar de que el conjunto de reglas de formación de siglas se considera universal, para cada lengua el orden de importancia de las reglas varía. Además, en ciertos casos puede haber reglas inaplicables, por ejemplo, la "concordancia de sílabas" no es aplicable en lenguas como el chino.

Zahariev establece el siguiente conjunto de reglas:

- *Concordancia inicial*. Concordancia de un caracter inicial de una palabra de la FD con un caracter equivalente en la sigla.

- *Concordancia de morfemas*. Concordancia de un caracter inicial de un morfema al interior de una palabra de la FD con un caracter equivalente en la sigla.
- *Concordancia de sílabas.* Concordancia de un caracter inicial de una sílaba al interior de una palabra de la FD con un caracter equivalente en la sigla.
- *Concordancia de grupos de caracteres*. Concordancia de un grupo de caracteres consecutivos en una palabra de la FD con un grupo equivalente de caracteres consecutivos en la sigla.
- *Concordancia de caracteres internos.* Concordancia de un caracter interno en una palabra de la FD con un caracter equivalente en la sigla.
- *Omisión de palabras gramaticales o stopwords*. Omisión en la sigla de una palabra gramatical presente en la FD.
- *Omisión de una palabra precedida por signos de puntuación.* Omisión en la sigla de una palabra precedida por ciertos signos de puntuación (guión y barra) presentes en la FD.
- *Omisión de una palabra.* Omisión en la sigla de un caracter que representa una de las palabras de la FD.
- *Formación del plural mediante duplicación*. Duplicación en la sigla de los caracteres equivalentes de una FD pluralizada .
- *Concordancia simbólica.* Concordancia de un símbolo, caracter, morfema, grupo de caracteres, palabra o expresión en la FD con un caracter o grupo de caracteres en la sigla, siguiendo reglas *ad-hoc*, las cuales son reconocibles generalmente por determinados grupos sociales (por ejemplo: CU por *"see you";* XMAS por *"Christmas"*).
- *Migración.* En las lenguas donde algunos acentos u otros signos pueden acompañar a los caracteres del alfabeto, los caracteres de la FD pueden "migrar" a la sigla como elementos no acentuados. Por ejemplo, en rumano "Î, Ş, Ţ" migran a "I, S, T", respectivamente. De igual modo, en francés la "É" migra a "E" como se comprueba en el siguiente caso: *"Électricité de France (EDF)"*.
- *Flexión.* En lenguas con morfología aglutinante, las concordancias de los grupos que representan morfemas enteros pueden flexionarse en la sigla.
- *Concordancia consecutiva.* Los caracteres de una sigla pueden concordar consecutivamente, en la misma dirección de los símbolos, caracteres o palabras

pertenecientes a la FD. Este es el principio que siguen algoritmos como *Longest common subsequence (LCS)*, empleado por Taghva & Gilbreth.

- *Inversión.* Hay situaciones en que los caracteres constitutivos de una sigla concuerdan con caracteres de la FD pero en orden inverso.
- *Préstamo.* Las siglas pueden prestarse directamente de otras lenguas en lugar de crearse a partir de la traducción de sus FD. Este caso es muy común en las siglas de áreas científico-técnicas. Algunos ejemplos son: HTTP, URL, DNA, etc.

En lo que respecta a la lengua española, Zahariev sostiene que las reglas predominantes en la formación de siglas son:

- *Concordancia inicial*; *e.g.*: TLCAN (Tratado de Libre Comercio de América del Norte)
- *Omisión de palabras gramaticales; e.g.*: SICAV (Sociedad de inversión de capital variable)
- *Concordancia de morfemas; e.g.*: SIDA (síndrome de inmunodeficiencia adquirida)
- *Formación del plural mediante duplicación; e.g.*: EEUU (Estados Unidos)
- *Concordancia de grupos de caracteres;* según el autor, esta regla se usa también para combinar nombres propios *e.g.*: Marisa (Maria Isabel).

b. Técnica de extracción de siglas

Para la detección de siglas Zahariev propone un algoritmo que opera en dos fases sucesivas, a saber: detección de pares sigla-FD y detección de FD. Los resultados de ambas fases se listan.

En la fase de detección de los pares sigla-FD, el proceso comienza con cada ocurrencia de una sigla. A partir de aquí el algoritmo busca el candidato a FD en el contexto que la rodea.

El método para hallar los candidatos a FD es similar al empleado por el algoritmo canónico-contextual de Larkey.

La búsqueda de la concordancia de letras entre la sigla y la FD se realiza mediante un conjunto de reglas flexibles que se aplican sucesivamente en diferente orden en el contexto que rodea al candidato a sigla. Las reglas son:

- *Concordancia inicial.* Cuando el caracter inicial de la palabra concuerda con el caracter equivalente en la sigla.
- *Omisión de palabras gramaticales.* Se aplica una lista de exclusión de palabras gramaticales (artículos, preposiciones y conjunciones).
- *Concordancia de subsiglas.* Cuando una sigla entera está incluida dentro de otra sigla; *e.g.*: VLAN por *"virtual LAN"*.
- *Concordancia de prefijo morfológico.* Se aplica una lista de prefijos del inglés para buscar concordancias con el prefijo de determinada palabra de la FD. Cuando hay coincidencia, tanto la inicial del prefijo como el resto de la palabra, se consideran parte de la sigla; *e.g.*: *"hypertext"* concuerda con "HT", el comienzo de "HTML", la sigla correspondiente a *"Hypertext Markup Language"*
- *Concordancia de prefijo.* Cuando las letras del comienzo de una palabra en la FD son idénticas al grupo de letras correspondiente en la sigla; *e.g.*: las primeras cuatro letras de la palabra *"bootstrap"* concuerdan, al comienzo de la sigla "BOOTP", sigla correspondiente a *"Bootstrap Protocol"*.
- *Concordancia orientada a la sílaba.* Cuando la o las primeras letras o consonantes en cada sílaba concuerdan con las letras correspondientes en la sigla; *e.g.*: la palabra *"connectionless"* se descompone en las sílabas *con-nec-ti-on-less*, y las iniciales de las sílabas *"con"* y *"less"* concuerdan con "CL" del comienzo de CLNP, una sigla para *"Connectionless Network Protocol"*.
- *Omisión de palabras.* Cuando las palabras introducidas por signos de puntuación como la barra o el guión se omiten en la concordancia de letras de la sigla; *e.g.*: la palabra *"Level"* se omite en la FD *"High-Level Data Link Control"*, correspondiente a la sigla HDLC.
- *X.* Cuando las concordancias incluyen la letra X dentro de una sigla, se considera que pueden concordar con la palabra "ex" al comienzo de palabras en la FD, como es el caso de "XML", la sigla correspondiente a *"Extensible Markup Language"*.

c. Evaluación

Zahariev evalúa el rendimiento de su algoritmo por medio de un corpus creado con los *abstracts* de la *Internet Engineering Task Force (IETF) Request for Comments (RFC).* El corpus de evaluación, conformado por 681 pares de candidatos sigla-FD, arrojó 98.58% de precisión y 93.19% de exhaustividad.

2.2 Automatic Acronym Identification and Creation of an Acronym Database

Young (2004) desarrolló una técnica de identificación de siglas y una base de datos para su almacenamiento. La investigación se centra en dos dominios: general (noticias) y especializado (biomedicina). En el primer caso emplea el sitio de la *BBC* y el corpus de la *agencia Reuters* mientras que en el segundo utiliza la base de datos de *Medline*.

Para la constitución del corpus de evaluación del sistema, Young creó los siguientes módulos:

- *Unidad de compilación de páginas web (Harvest unit).* Busca y reúne para su procesamiento todos los documentos útiles provenientes de las páginas web.
- *Unidad de filtro para remoción de las etiquetas HTML.* Elimina todas las etiquetas HTML de los documentos, dejándolos en texto plano.

a. Patrones

Young basa su trabajo en el vector de rasgos sugerido por Chang *et al*. (2002), el cual se describe a continuación:

- *Minúscula vs. mayúscula*. Gran parte de las de siglas contienen más letras mayúsculas que minúsculas.
- *Comienzo de palabra*. Los listados de siglas existentes muestran que las siglas se crean, generalmente, a partir de las primeras letras de las palabras de la FD.
- *Final de palabra*. Existen otros sitios en la FD de los cuales se puede tomar letras como son los finales de las palabras.

- *Límite de sílaba*. Es un sitio lógico para escoger letras que formen una sigla, en especial cuando el límite de la sílaba es el límite entre dos palabras.
- *Después de letra alineada*. Las letras que siguen a la letra previa escogida también pueden hacer parte de la sigla.
- *Letras alineadas*. Si hay un alto número de letras no alineadas es probable que el candidato a FD no sea el correcto.
- *Palabras omitidas*. Es el número de palabras de la FD que no concuerdan con ninguna letra de la sigla. El vector de rasgos utilizado descartará cualquier palabra vacía (o *stopword*) en el cálculo de este rasgo.
- *Letras alineadas por palabra*. Las listas de siglas muestran que generalmente se alinea una letra de la sigla por una palabra de la FD. Un gran número de letras alineadas por palabra muestra que se toman demasiadas letras de palabras individuales, lo cual indica una concordancia adicional.

Aparte de los rasgos establecidos por Chang, Young sugiere la inclusión de un rasgo adicional: la clasificación (*ranking*) por medio de internet. Para calcular dicha clasificación se emplea un buscador. El cálculo se basa en el porcentaje de páginas halladas para todos los candidatos sigla-FD. Esta clasificación es útil puesto que usa internet como un sistema de votación; cada página que presenta el par sigla-FD cuenta como un voto para la elección de la FD correcta.

b. Técnica de extracción de siglas

El extractor identifica los candidatos a sigla mediante el uso de la siguiente expresión regular:

[a-zA-Z0-9] * [([A-Z] {2}) ([A-Z] [0-9]) ([0-9] [A-Z])] [a-zA-Z0-9]*

Esta se aplica a una lista de palabras *tokenizadas* por espacios en blanco. La parte central de la expresión, *i.e.*: [([A-Z] {2}) ([A-Z] [0-9]) ([0-9] [A-Z])] indica que una porción del texto debe tener bien dos letras mayúsculas [([A-Z] {2}), una letra

mayúscula y un número ([A-Z] [0-9]), o bien un número y una letra mayúscula ([0-9] [A-Z]).

El sistema analiza la ventana de texto que hay a la derecha e izquierda del candidato a sigla, sin importar si hay o no mayúsculas. Cuando se encuentra una sigla se registra su posición y se extrae junto con su ventana de texto para pasarla luego por el módulo de deducción de la FD.

La extensión de la ventana de texto varía de acuerdo con la longitud de la sigla. Si la sigla contiene menos de 7 caracteres, la extensión equivaldrá al número de letras de la sigla multiplicado por 2. Si la sigla contiene más de 7 caracteres, se multiplicará por 1.2 con el fin de limitar el número de posibles candidatos a FD. En particular, las siglas extensas producen ventanas de texto extensas lo que significa que podrían inferirse más pares de candidatos sigla-FD.

Para la extracción de las FD el sistema sigue los siguientes pasos:

- *Identificación de la FD.* El módulo de extracción de las FD infiere los candidatos a partir del contexto donde aparece la sigla. Este módulo emplea el algoritmo *Longest common subsequence* (LCS), por su facilidad para generar todos los candidatos sigla-FD sin necesidad de usar heurística *ad-hoc*. Los pares sigla-FD se envían posteriormente al módulo de clasificación (*ranking*).
- *Clasificación (ranking).* El módulo de clasificación asigna un puntaje a los pares sigla-FD. De esta manera, el par que tenga el mayor puntaje se convertirá en el correcto. Este módulo se basa en un algoritmo de aprendizaje máquina supervisado. El factor clave en el éxito de este tipo de algoritmo radica en la selección de un vector de rasgos que describe la estructura de la sigla y su contexto.

Para el almacenamiento de los pares sigla-FD, Young ha creado una interfaz de internet compuesta por la BD de siglas y el sitio web.[14]

[14] Actualmente no se encuentra disponible para su consulta

La BD almacena todos los pares sigla-FD correctos junto con el porcentaje de probabilidad de exactitud.

La interfaz permite tanto la consulta de la BD de siglas como la sugerencia de nuevas siglas por parte de los usuarios.

En síntesis, Young presenta un diseño de sistema modular, con el que logra que cada sección pueda ejecutarse independientemente. La arquitectura general del sistema es la siguiente:

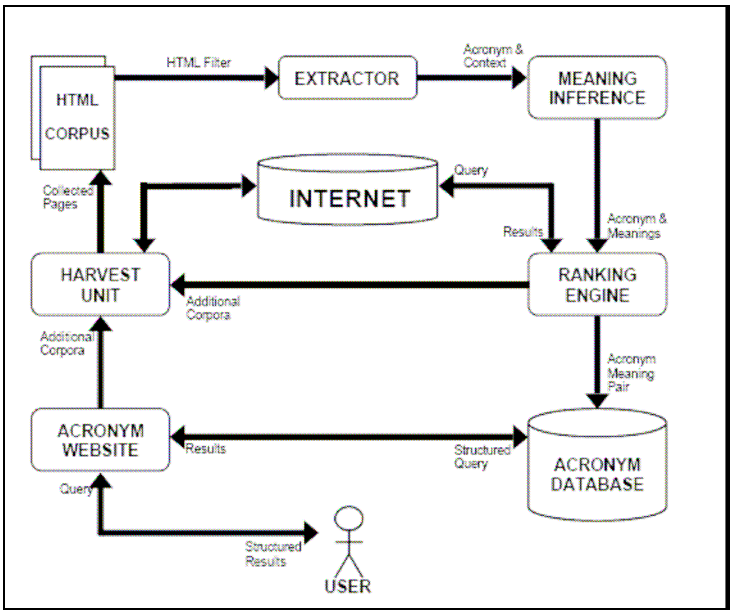

Fig. 8. Aspecto general del sistema de extracción y generación de la base de datos de siglas (Young, 2004)

c. Evaluación

Para la evaluación del sistema, Young tuvo en cuenta dos casos. En el primero, el sistema procesó artículos del corpus de *Reuters* seleccionados aleatoriamente. En el segundo, el sistema procesó el *Gold Standard* de *Medstract* con el fin de realizar comparaciones directas con los resultados de otros trabajos.

1) Evaluación del corpus de *Reuters*

El sistema procesó 100 artículos procedentes del corpus de *Reuters* dando como resultado 94% de precisión y 77% de exhaustividad. No obstante, esta autora advierte

que, de momento, estos resultados no son comparables ya que no se cuenta con otros trabajos que hayan empleado corpus de noticias.

2) Evaluación contra el *Gold Standard* de *Medstract*

Young evaluó su sistema con el "*Gold Standard*" de *Medstract* para poder establecer comparaciones con los resultados de otros trabajos.[15] Sin embargo, por limitaciones de tiempo, no se pudo entrenar el sistema que ejecuta el *Gold Standard* de *Medstract* con textos biomédicos sino con artículos noticiosos.

Teniendo en cuenta esta limitación, el sistema de Young logró identificar correctamente 47 pares y marcó otros 5 pares como correctos cuando en realidad se trataba de falsos positivos. El sistema no detectó 38 pares y el resto los descartó bien porque se trataba de siglas duplicadas o porque no se ajustaban a la definición de sigla establecida para este trabajo.

El sistema de Young obtuvo 92.2% de precisión y 55.3% de exhaustividad; porcentajes que, según la autora, son favorables si se considera que su sistema no pudo entrenarse con literatura biomédica.

2.3 A supervised learning approach

Nadeau & Turney (2005) proponen un sistema de detección de siglas basado en aprendizaje supervisado.

a. Heurística

Un candidato a sigla es un *token* conformado por *1-n* caracteres alfanuméricos, los cuales pueden incluir puntos. La primera letra de la FD debe coincidir con la primera letra de la sigla. La FD no debe contener signos de puntuación tales como: [] , ; : ¿? ¡!

El sistema emplea 17 rasgos identificables en un candidato a sigla-FD, como son:

- Número de letras que coinciden con la primera letra de una palabra de la FD

[15] El *Gold Standard* de *Medstract* contiene 168 pares de siglas-FD.

- La heurística anterior se basa en la longitud de la sigla
- Número de letras de la FD que están en mayúscula
- La heurística anterior se basa en la longitud de la sigla
- Longitud (en palabras) de la FD
- Distancia (en palabras) entre la sigla y la FD
- Número de palabras de la FD que no participan
- La heurística anterior se basa en la longitud de la FD
- Tamaño medio de las palabras en la FD que no participan
- Categoría gramatical de la primera palabra de la FD (es decir, si es preposición, conjunción o determinante)
- Categoría gramatical de la última palabra de la FD (es decir, si es una preposición, conjunción o determinante)
- Número de preposiciones, conjunciones o determinantes en la FD
- Número máximo de letras que participan en una sola palabra de la FD
- Número de letras de la sigla que no participan
- Número de dígitos de la sigla y puntos que no participan
- Presencia de paréntesis en la sigla o en la FD
- Número de verbos en la FD.

b. Técnica de extracción

El sistema emplea un conjunto de heurísticas tanto para identificar los candidatos a FD como para limitar el contexto (ventana) de búsqueda. Luego, los candidatos a sigla se convierten en vectores. Cada vector consiste en 17 características que describen cada miembro de los pares de siglas.

Para determinar si un par sigla-FD es correcto, el algoritmo lo confronta con un corpus anotado. Un par sigla-FD se etiqueta como válido si hay una coincidencia exacta entre la sigla y la FD en el corpus.

c. Evaluación

Una heurística flexible permite la identificación de un gran número de candidatos sigla-FD lo cual se traduce en un alto grado de exhaustividad. Estos autores encontraron que

los rasgos más productivos en estricto orden son: 1) distancia entre la FD y la sigla; 2) número de letras de la sigla que concuerdan con las primeras letras de las palabras de la FD; y 3) uso de paréntesis.

El algoritmo empleado es el *SVM (Support Vector Machine) SMO* de *WEKA*, el cual se probó contra el *Gold Standard* de *Medstract*, usado como corpus de evaluación. Con esta prueba se detectaron correctamente 126 pares de sigla-FD de un total de 168, lo que implica en términos de rendimiento 92.5% de precisión y 88.4% de exhaustividad.

2.4 Recognizing acronyms in Swedish texts

Dannélls (2006) implementó un sistema para el reconocimiento de siglas en textos de biomedicina escritos en sueco. El sistema usa una heurística general para identificar y extraer los candidatos sigla-FD. Posteriormente, un algoritmo de aprendizaje máquina se encarga de clasificar estos candidatos. Y, por último, los pares clasificados correctamente se almacenan en una base de datos.

a. Heurística

Un candidato a sigla es una cadena de caracteres alfabéticos, numéricos y especiales (guiones o barras).

Un candidato se considera válido cuando cumple con las condiciones 1 y 2 y 3 ó 4, que se enumeran a continuación:

1) contener al menos dos caracteres
2) no pertenecer a la lista de palabras rechazadas (*stopwords*)
3) contener al menos una letra mayúscula
4) poseer como caracter final una minúscula o un número.

b. Técnica de extracción de siglas

El método empleado por Dannélls es similar al algoritmo utilizado por Schwartz & Hearst, pero con la ventaja de que puede reconocer pares de sigla-FD que no están marcados por paréntesis.

Cuando se encuentra una sigla, el algoritmo busca su FD correspondiente en las palabras aledañas. El candidato a FD debe cumplir con todas y cada una de las siguientes condiciones:

1) que al menos una letra de las palabras concuerde con una letra en la sigla
2) que la cadena de caracteres de la sigla no contenga signos como: ";", ":", "?", "!"
3) que la longitud máxima de la cadena de caracteres sea min(|A|+5, |A|*2), donde |A| es la longitud de la sigla
4) que la cadena de caracteres no contenga solo letras mayúsculas.

De acuerdo con estas condiciones, el proceso de extracción de pares sigla-FD consta de dos fases, a saber:

1) *Concordancia de paréntesis.* En la práctica la mayoría de los pares sigla-FD se ajustan a alguno de estos patrones: "FD (sigla)" o "sigla (FD)". Por lo tanto, el algoritmo extrae los candidatos sigla-FD que cumplen con esta condición.
2) *Ausencia de concordancia de paréntesis.* El algoritmo busca candidatos a sigla que cumplan las cuatro condiciones antes mencionadas y que no se encuentren entre paréntesis. Una vez se detecta un candidato a sigla, el algoritmo rastrea el contexto (ventana) anterior y posterior en busca de un candidato a FD. El tamaño de la ventana corresponde al resultado de multiplicar cuatro palabras por el número de letras del candidato a sigla.

La selección del candidato sigla-FD correcto se hace mediante la reducción de la ventana donde aparece el candidato a FD, así:

- El algoritmo busca caracteres idénticos entre el candidato a sigla y el candidato a FD comenzando desde el final de ambas cadenas de caracteres. El candidato par sigla-FD es correcto si satisface las siguientes condiciones:

    - que al menos un caracter de la sigla concuerde con un caracter de la FD

- que el primer caracter en la sigla concuerde con el primer caracter de la primera palabra de la FD, independientemente de que esté en mayúscula o minúscula.

Dannélls emplea un algoritmo de aprendizaje máquina, el cual requiere que los candidatos sigla-FD se representen como vectores de rasgos. La selección de estos rasgos es importante tanto para el proceso de aprendizaje como para la selección del algoritmo y método de entrenamiento del clasificador.

El cálculo de los vectores de rasgos para describir los pares sigla-FD se basa en los diez rasgos siguientes:

1) la sigla o la FD están entre paréntesis (0 falso, 1 verdadero)
2) la FD aparece antes de la sigla (0 falso, 1 verdadero)
3) la distancia en palabras *(offset)* entre la sigla y la FD
4) el número de caracteres de la sigla
5) el número de caracteres de la FD
6) el número de minúsculas en la sigla
7) el número de minúsculas en la FD
8) el número de mayúsculas en la sigla
9) el número de mayúsculas en la FD
10) el número de palabras en la FD.

Además, Dannélls menciona un rasgo adicional, relacionado con el tipo de predicción; *i.e.*: candidato verdadero (+), candidato falso (-).

La siguiente es una representación de un par sigla-FD mediante un vector de rasgos:

Sigla-FD	**Vector de rasgos**										
	Rasgo1	Rasgo2	Rasgo3	Rasgo4	Rasgo5	Rasgo6	Rasgo7	Rasgo8	Rasgo9	Rasgo10	Rasgo11
VCJD-variant CJD	0	0	1	4	11	1	7	3	3	2	+

c. Evaluación

El corpus para evaluar el sistema consta de 861 pares sigla-FD, extraídos del corpus MEDLEX (textos de medicina en sueco). Dicho corpus se anotó manualmente con etiquetas XML.

El algoritmo detectó 671 pares, de los cuales 47 eran incorrectos, lo que supone 93% de precisión y 72.5% de exhaustividad. El sistema detectó erróneamente 47 porque:

- las palabras que aparecen en la FD no tienen una letra correspondiente en la sigla
- las letras en la sigla no tienen una palabra correspondiente en la FD; *e.g.*: *"PGA, glycol alginate lösning"*
- los caracteres en la FD no concuerdan con los caracteres de la sigla.

El análisis de errores mostró las causas por las que el sistema no identificó los 190 pares sigla-FD restantes. Estas son:

- las letras de la FD no aparecen en la sigla (esto se debe básicamente a que la FD aparece traducida al sueco mientras que la sigla se mantiene en inglés)
- la mezcla de números arábigos con romanos, *e.g.*: *"USH3, Usher type III"*
- la posición de números/letras
- las siglas de tres caracteres que aparecen en minúsculas.

El algoritmo de aprendizaje máquina de mejor resultado es el IB1, pues clasificó correctamente el 98.8% de los pares sigla-FD.

A modo de síntesis, se presenta la siguiente tabla comparativa del rendimiento de los sistemas de detección de siglas basados en métodos estadísticos y de aprendizaje máquina.

Sistema	Método	Autor	Año	Corpus	Área	Precisión	Exhausti-vidad
Dicc. en línea de abreviaciones	Estadístico	Chang *et. al*	2002	*Gold Standard* de *Medstract* (168 siglas)	Biomed.	80%	83%
SaRAD	Estadístico	Adar	2002	*Gold Standard* de *Medsrtact* (168 siglas)	Biomed.	86%	88%
s.n.	Aprendizaje máquina	Zahariev	2004	*Abstracts* de *Internet Engineering Task Force-Request for comments* (681 siglas)	Internet-Informática	99%	93%
s.n.	Aprendizaje máquina	Young	2004	100 artículos corpus agencia *Reuters*	General	94%	77%
				Gold Standard de *Medstract* (168 siglas)	Biomed.	92%	55%
s.n.	Aprendizaje máquina	Nadeau & Turney	2005	*Gold Standard* de *Medstract* (168 siglas)	Biomed.	92%	88%
s.n.	Aprendizaje máquina	Dannélls	2006	Corpus de textos en sueco *"MEDLEX"* (861 siglas)	Medicina	93%	72%

Tabla 9. Rendimiento de los sistemas de detección de siglas basados en métodos estadísticos y de aprendizaje máquina

De los datos recogidos en la tabla anterior se desprende que el sistema basado en estadística con mejor desempeño es el de Adar. Así mismo, los sistemas basados en aprendizaje máquina con mejor rendimiento son los de Zahariev y Nadeau & Turney.

3. Métodos híbridos

Park & Byrd (2001) emplean un método basado en tres tipos de conocimiento: reglas de formación de siglas, marcadores textuales y palabras clasificadoras.

Las reglas de formación de siglas describen cómo se forma una sigla a partir de su FD. Los marcadores textuales son símbolos especiales que se usan para indicar la relación de siglas y FD en los textos; *e.g.*: "()", "[]" o "=". Y las palabras clasificadoras indican una fuerte relación entre la sigla y su FD; *e.g.*: *"or", "short", "acronym", "stand"*, etc.

El sistema realiza cinco procesos, a saber:

1) detección de siglas
2) búsqueda de FD

3) aplicación de reglas
4) concordancia de siglas
5) selección del mejor candidato a sigla-FD.

a. Patrones

Park & Byrd consideran como candidato a sigla aquella cadena de caracteres que cumple con las siguientes condiciones:

- primer caracter alfanumérico
- longitud entre 2 y 10 caracteres
- mínimo una letra mayúscula

Y se ajusta a las siguientes restricciones:

- no es una palabra recogida en un diccionario ni aparece como la primera palabra de una oración
- no es nombre de persona o lugar
- no es a una *stopword*.

b. Técnica de extracción de siglas

Cuando se encuentra un candidato a sigla, el algoritmo determina el contexto de búsqueda de la FD, el cual tiene una longitud máxima de +10 palabras a la derecha e izquierda del candidato a sigla.

Cuando se encuentra un par sigla-FD, se generan los patrones que describen la sigla y la FD, así:

1) patrones para la sigla
- los caracteres alfabéticos se reemplazan con una “c”
- los caracteres numéricos se reemplazan con una “n”.

Por ejemplo:

Sigla	Patrones
2MASS	Ncccc
NEXT	Cccc
R&D	Cc
SN1987A	Ccnc

2) patrones para la FD

- *word* (w)
- *stopword* (s)
- *prefix* (p)
- *headword* (h)
- *number* (n).

Por ejemplo:

FD	patrones
Supernova 1987A	phnw
Two-Micron All Sky Survey	wwwww
U.S. Department of Agriculture	wwsw

Teniendo en cuenta lo anterior, los patrones para el par sigla-FD *"X2B (Hexadecimal to Binary"* son: (“cnc”, “phsw”).

Posteriormente, se generan las reglas de siglas, las cuales se usan para describir cómo se forma una sigla a partir de su FD. Una regla de formación consiste en: un patrón de sigla, un patrón de FD y una regla de formación.

Una regla de formación define cómo se forma cada caracter de una sigla a partir de su FD. Un elemento en una regla de formación tiene dos valores; *i.e.*: un número de palabra (ubicación de la palabra dentro de la FD) y un método de formación (existen 5 métodos de formación).

Los métodos de formación son: “F” (primer caracter), “I” (caracter interior), “L” (último caracter), “E” (caracter exacto, para caracteres numéricos) y “R” (reemplazo de concordancia).

El sistema cuenta inicialmente con una base de 45 reglas de formación de siglas, extraídas a partir del análisis de un corpus de 4.500 siglas del ámbito de la informática.

A continuación se muestran dos ejemplos de reglas de formación de siglas.

	Par sigla-FD
Regla de formación	2-MASS Two-micron All Sky Survey <ncccc, wwwww, (1,R) (2,F) (3,F) (4,F) (5,F)>
Regla de formación	CONTOUR Comet Nuclear Tour <ccccccc, www, (1,F) (1,I) (2,F) (3,F), (3,I) (3,I) (3,L)>

c. Evaluación

El sistema se probó en tres corpus diferentes, ingeniería automotriz, farmacéutica y boletines de prensa de la NASA. El sistema no detectó algunas siglas, básicamente porque:

- las FD se encontraban fuera del contexto de búsqueda establecido
- el etiquetador de POS hizo una mala interpretación.

La siguiente tabla indica el rendimiento de este sistema en cada uno de los corpus.

Sistema	Método	Autor	Año	Corpus	Área	Precisión	Exhausti-vidad
s.n.	Híbrido	Park & Byrd	2001	20.379 palabras (33 siglas)	Ing. automotriz	96.9%	93.9%
				97.000 palabras (63 siglas)	Farmacéutica	100%	95.2%
				83.539 palabras (81 siglas)	Boletines prensa de la NASA	97.4%	93.8%

Tabla 10. Rendimiento del sistema (Park & Byrd, 2001)

El análisis de los datos de esta tabla muestra que los resultados del sistema son bastante parecidos a los obtenidos por los métodos de aprendizaje máquina de Zahariev, mencionados en el apartado anterior.

II. Sistemas de desambiguación de siglas

Se entiende por desambiguación de siglas al mecanismo de selección de la FD apropiada para una ocurrencia específica de una sigla en un contexto dado. Por ejemplo, si se pretende recuperar documentos relacionados con *SRF*, con el sentido de "*Serum Response Factor*", no deberían recuperarse aquellos documentos que contengan la cadena *SRF* con un significado diferente como *"Spatial Receptive Field"*. A menudo, no es posible desambiguar el sentido de la sigla por medio de expresiones buleanas porque simplemente no existe la FD en el documento.

Diferentes estudios llevados a cabo por Liu *et al.* (2001, 2002) muestran que el 33% de las siglas listadas en el *Unified Medical Language System (UMLS)* en 2001 son ambiguas. En un estudio posterior, estos autores demostraron que el 81% de las siglas encontradas en los *abstracts* de *Medline* eran ambiguas y tenían 16 sentidos en promedio.

Entre los autores que han desarrollado sistemas de desambiguación se encuentran: Pustejovsky (2001), Pakhomov (2002), Yu (2003), Adar (2004), Zahariev (2004), Bracewell *et al.* (2005), Gaudan *et al.* (2005) y Joshi *et al.* (2006).

En general, los métodos de desambiguación de siglas realizan el siguiente proceso:

- uso de un lexicón para la compilación de las siglas y sus sentidos (o FD)
- cómputo del contexto de uso para cada sentido
- entrenamiento de un algoritmo de aprendizaje máquina con el contexto de cada sentido.

Estos pasos pueden observarse en la siguiente gráfica de Gaudan *et al.* (2005).

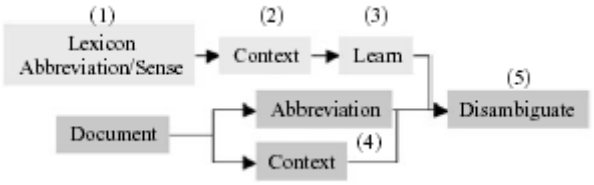

Fig. 9.

A continuación se describen dos de los sistemas de desambiguación con los que se cuenta en la actualidad.

A. Polyfind

Pustejovsky *et al.* (2001) desarrollaron el algoritmo para la desambiguación de siglas *Polyfind*. Para la evaluación de este se escogió la sigla *SRF* con diez FD diferentes y se recogieron todos los *abstracts* en *Medline* que contuvieran estos pares sigla-FD. Los *abstracts* se agruparon de acuerdo con la FD que contenían, resultando diez grupos, que servirían como plantillas de documento contra las cuales se evaluarían las ocurrencias ambiguas de *SRF*. Posteriormente, se recogieron 42 *abstracts* en los cuales esta sigla aparecía sin la FD. Estas ocurrencias de *SRF* se desambiguaron manualmente y se dividieron en cuatro grupos, de acuerdo con la FD a la que correspondían: *Serum Response Factor*, *Subretinal Fluid*, *Surfactin* y *C Elegans surface antigen gene mutations.*

Los *abstracts* de cada uno de estos cuatro grupos se usaron más adelante como consultas, y se evaluaron contra los diez conjuntos de FD. Tanto la búsqueda como los vectores de la plantilla del documento incluían los valores de los *tokens* del título, los nombres de los autores, el título del *journal* y el cuerpo del *abstract*.

Pustejovsky *et al.* emplearon como estándar el método de puntuación "atc". Posteriormente, se computó la medida de similitud entre la búsqueda y cada uno de los conjuntos de FD. La consulta se consideraba desambiguada correctamente si el conjunto con la FD correcta de SRF obtenía el mayor puntaje de similitud.

En definitiva, los resultados preliminares de *Polyfind* demuestran que la aplicación de un modelo de vector espacial para la desambiguación de los sentidos de siglas polisémicas es prometedor, *Polyfind* alcanzó 97.2% de exactitud en la desambiguación.

B. Automatic resolution of ambiguous abbreviations in Biomedical texts

Yu *et al.* (2003) han implementado un algoritmo para desambiguar las siglas en los *abstracts* de la base de datos *Medline*. Este sistema se basa en el uso de máquinas de soporte vectorial (SVM) y en la hipótesis de que "todas las ocurrencias de una sigla dentro de un *abstract* tienen la misma FD". En la evaluación, este sistema alcanzó una exactitud de 87%.

Las SVM emplean como corpus de entrenamiento un corpus etiquetado. Para las tareas de desambiguación, el corpus contiene vectores, donde cada vector es una descripción de la ocurrencia de una abreviación. Dicho vector tiene la forma de rasgos *(feature$_1$, feature$_2$....feature$_n$, label)*, donde *label* representa cuál FD está siendo usada en una ocurrencia de una sigla y *feature$_1$, feature$_2$.... feature$_n$* describen el contexto donde aparece la sigla; es decir, las dos palabras a la derecha e izquierda de cada ocurrencia de la sigla. Yu *et al.* lo ilustran así:

> "OBJECTIVES: The aim of the present study was to assess the **contribution of** angiotensin-converting enzyme (ACE) **inhibiror therapy** to bradykinin-induced tissue type plasminogen activator (t-PA) release in patients with heart failure (HF) secundary to ischemic heart disease. BACKGROUND: Bradykinin is a potent endothelial cell stimulant that causes vasodilatation and t-PA release. In large-scale clinical **trials,** ACE **inhibitor therapy** prevents ischemic events....".

Los vectores extraídos de este fragmento de *abstract* para la FD *"angiotensin-converting enzyme"* de la sigla *"ACE"* son los siguientes:

L2= contribution, L1=of, R1=inhibitor, R2=therapy
L2=trials, L1= ",", R1=inhibitor, R2=therapy

En síntesis, El método de Yu *et al.* consiste en:

- emplear el algoritmo SVM, de clasificación supervisada, para predecir la FD probable de una sigla
- extraer mediante SVM los datos usando las FD de las siglas desambiguadas
- utilizar la hipótesis " un sentido por documento".

El algoritmo propuesto es el siguiente:

Input:

(1) A set of Medline abstracts (SMA)

(2) Abbreviation dictionary containing the abbreviations and their long forms disambiguated (AD)

Output:

Set of vectors in form (Feature$_1$, Feature$_2$, …, Feature$_n$, A, LF), where A is an abbreviation, and LF is one of the long forms of the abbreviation A (every vector represents an occurrence of the abbreviation A in context (Feature$_1$, Feature$_2$, …, Feature$_n$,) and the abbreviation A in the context has the sense (long form) LF)

Algorithm:

```
FOR (X ∈SMA) DO   //X is an abstract in the set SMA
{  FOR (A ∈AD) DO
  //A is an abbreviation in the dictionary AD
   { IF ((a long forms of abbreviation A is found in X)
       AND  (the found long form is LF)
       AND (no other long form of A is also found in X))
     {   FOR (each of the occurrences of A in X) DO
       {  Generate vector (Feature1, Feature2, …, Featuren,
                                A, LF),
          where Feature1, Feature2, …, Featuren, is the
           context of the occurrence;
       }
     }
}
```

Fig. 10. Algoritmo empleado (Yu et al., 2003)

III. Criterios para el diseño de un modelo de detector-extractor de siglas para el español

El primer paso que se debe cumplir durante el diseño de un sistema de adquisición de siglas consiste en determinar las reglas que rigen estas unidades; es decir, la"heurística". A partir del análisis de los trabajos citados anteriormente, se pueden deducir las reglas de formación y de concordancia, así como los patrones de identificación de las siglas.

A. Reglas de formación de siglas

Partiendo de la idea de que una regla es el modo en que se produce algo, puede decirse que existen 2 tipos de reglas de formación de siglas: básicas y complementarias.

1. Reglas básicas de formación de siglas

a. Mínimo 2 caracteres y máximo 10
b. Máximo 2 palabras
c. Mínimo una letra mayúscula
d. Primer caracter alfanumérico
e. Exclusión de los signos ; : ? !

2. Reglas complementarias de formación de siglas

a. *Inclusión*. Una sigla puede incluir signos como barras, puntos o guiones
b. *Plural*. Para formar el plural de la sigla puede añadirse una "s", caso recurrente en siglas creadas en inglés como *retinoid x receptors (RXRs)*; o duplicarse los caracteres de la sigla, caso propio de lenguas como el español; *e.g.*: EEUU.
c. *Caracteres alfanuméricos*. El primer o último caracter de la sigla puede ser una letra o un número.

B. Reglas de concordancia pares sigla-FD

1. Concordancia de caracteres

a. *Concordancia inicial*. Concordancia de un caracter inicial de una palabra de la FD con un caracter equivalente en la sigla.
b. *Concordancia de morfemas*. Concordancia de un caracter inicial de un morfema al interior de una palabra de la FD con un caracter equivalente en la sigla.

c. *Concordancia de sílabas*. Concordancia de un caracter inicial de una sílaba al interior de una palabra de la FD con un caracter equivalente en la sigla. Es decir, la primera letra(s) o sílaba(s) en cada sílaba concuerda(n) con las letras equivalentes de la sigla; *e.g.*: la palabra *"connectionless"* se descompone en las sílabas *con-nec-ti-on-less*, y las iniciales de las sílabas *"con"* y *"less"* concuerdan con "CL" del comienzo de CLNP, la sigla de *"Connectionless Network Protocol"*.

d. *Concordancia de grupos de caracteres*. Concordancia de un grupo de caracteres consecutivos en una palabra de la FD con un grupo equivalente de caracteres consecutivos en la sigla.

e. *Concordancia de caracteres internos*. Concordancia de un caracter interno en una palabra de la FD con un caracter equivalente en la sigla.

f. *Concordancia simbólica*. Concordancia de un símbolo, caracter, morfema, grupo de caracteres, palabra o expresión en la FD con un caracter o grupo de caracteres en la sigla, siguiendo reglas *ad-hoc*, las cuales son generalmente reconocibles por determinados grupos sociales (por ejemplo, CU por *"see you";* XMAS por *"Christmas"*).

g. *Concordancia consecutiva*. Concordancia de caracteres de la sigla en la misma dirección de los símbolos, caracteres o palabras pertenecientes a la FD. Este es el principio que siguen algoritmos como *Longest common subsequence (LCS)*, empleado por Taghva & Gilbreth.

h. *Concordancia de subsiglas*. Una sigla entera puede estar incluida dentro de otra sigla; *e.g.*: *VLAN* por *"virtual LAN"*.

i. *Concordancia de prefijo morfológico*. Se aplica una lista de prefijos del inglés para buscar concordancias con el prefijo de determinada palabra en la FD. Cuando hay coincidencia, tanto la inicial del prefijo como el resto de la palabra se consideran parte de la sigla; *e.g.*: *"hypertext"* concuerda con "HT", el comienzo de "HTML", la sigla correspondiente a *"Hypertext Markup Language"*.

2. Inversión. Hay situaciones en que los caracteres constitutitos de una sigla concuerdan con caracteres de la FD pero en orden inverso.

3. Inserción. Una palabra está presente en la FD de una sigla, pero no ha sido usada en la formación de la sigla; *e.g.*: *thyroid hormone receptor (TR).*

4. Omisión. Existen tres casos típicos de omisión, a saber:

a. *Omisión de palabra.* Una palabra inexistente en la FD de una sigla se usa al momento de formar la sigla; *e.g.*: *[human] estrogen receptor (hER).*
b. *Omisión de palabra gramatical.* Un artículo, preposición o conjunción presente en la FD puede desaparecer al formar la sigla; *e.g.*: *VIH (Virus de la inmunodeficiencia humana).*
c. *Omisión de palabras separadas por signos de puntuación.* Palabras introducidas por signos de puntuación como la barra o el guión pueden omitirse, *e.g.*: la palabra *"Pacific"* presente en *"Asia-Pacific Association for Machine Translation"* se omite en su sigla *AAMT*.

5. **Sigla recursiva.** La FD de una sigla contiene a su vez otra sigla o abreviatura; *e.g.*: *CREB-binding protein (CBP).*

C. Patrones para la identificación de pares sigla-FD

Un patrón es aquella cosa que se toma como punto de referencia para valorar otras cosas de la misma especie. En este sentido, el presente trabajo considera dos tipos de patrones, a saber: patrones para identificación de candidatos a sigla y patrones para identificación de candidatos pares sigla-FD.

1. Patrones para la identificación de candidatos a sigla

Con base en el análisis de los trabajos reseñados aquí, en especial en Larkey *et al.* (2000), y en el análisis de nuestro corpus, se establecen los siguientes patrones de identificación de siglas.

- (U {sep})2-9S

U= *Uppercase* o mayúscula
{sep}= punto o punto seguido por un espacio
2-9= rango de caracteres
S= marca de plural (No todas las siglas la contienen. Es muy frecuente en las siglas creadas en inglés).

Una sigla puede presentar entre 2 y 9 caracteres en mayúscula, puede contener puntos y puede emplear la marca de plural, *e.g.*: U.S.A, U.S.A.'s.

- U2-9S

U= mayúscula
2-9= rango de caracteres
S= marca de plural (No todas las siglas la contienen. Es muy frecuente en las siglas creadas en inglés).

Una sigla puede presentar entre 2 y 9 caracteres e ir acompañada de la marca de plural; *e.g.*: USA, USA's.

- U*{dig}U+

U= mayúscula
*= 0 ó más ocurrencias
{dig}= número entre 1 y 9, opcionalmente seguido de un guión
+ = una o más ocurrencias de un caracter.

Una sigla puede presentar cero o más ocurrencias de caracteres en mayúscula seguidos de un número entre 1 y 9 y una ó más ocurrencias de caracteres en mayúscula; *e.g.*: 3D, 3-D, I3R.

- U+L+U+

U= mayúscula
+ = una ó más ocurrencias de un caracter
L= *Lowercase* o minúscula.

Una sigla puede contener uno ó más caracteres en mayúscula seguidos de uno ó más caracteres en minúscula seguidos de uno ó más caracteres en mayúscula; *e.g.*: DoD.

- U+[/-]U+

U= mayúscula
+ = una ó más ocurrencias de un caracter
[/-]= caracter separador barra o guión.

Una sigla puede estar formada por uno ó más caracteres en mayúscula separados por un guión o barra seguido de uno ó más caracteres en mayúscula; *e.g.*: AFL-CIO.

2. Patrones para la identificación de pares sigla-FD hallados en los trabajos analizados

Para establecer los patrones más frecuentes para la identificación de pares de candidatos sigla-FD, partimos del análisis de los trabajos reseñados. La mayoría de los autores, concretamente, Larkey *et al.* (2000), Pustejovsky *et al.* (2001), Schwartz & Hearst (2003), Nenadić *et al.* (2002), Adar (2004), Nadeau & Turney (2005) y Dannélls (2006), coincide en que el patrón más frecuente es **FD (SIGLA)**, hecho que igualmente hemos constatado en nuestro corpus. Los patrones encontrados son los siguientes:[16]

1) FD (SIGLA)
2) FD, SIGLA,
3) SIGLA (FD)
4) FD, SIGLA.
5) SIGLA, FD,
6) "FD" (SIGLA)
7) SIGLA = FD
8) SIGLA – FD
9) (FD) SIGLA
10) (SIGLA) FD
11) SIGLA or FD
12) FD or SIGLA
13) SIGLA stands for FD
14) SIGLA is an acronym for
15) FD known as the SIGLA
16) FD "SIGLA"
17) "SIGLA" FD

3. Patrones para la identificación de pares sigla-FD hallados en el corpus de este estudio[17]

El análisis de nuestro corpus ha permitido identificar los siguientes patrones:

#	Patrón	GH	MA	Presente en trabajos de otros autores
		Frecuencia	Frecuencia	
1	FD (SIGLA)	295	192	Sí
2	SIGLA (FD)	102	52	Sí
3	SIGLA ("FD")	13	1	No
4	SIGLA, FD	12	6	No

[16] Es de notar que estos autores han hallado estos patrones en corpus de textos en inglés.
[17] El corpus está constituido por 831 siglas en Genoma humano y 312 en Medio ambiente.

5	"FD" (SIGLA)	9	8	Sí
6	(SIGLA, FD)	9	2	No
7	FD o SIGLA	8	-	Sí
8	FD (SIGLA,)	6	-	No
9	(FD, SIGLA)	5	1	No
10	FD, SIGLA	4	4	No
11	(SIGLA, del inglés FD)	3	-	No
12	(SIGLA; FD)	3	-	No
13	FD, o SIGLA,	3	-	No
14	SIGLA, o FD	3	-	No
15	("FD" o SIGLA)	2	-	No
16	(SIGLA o FD)	2	-	No
17	FD (abreviado, SIGLA)	2	-	No
18	La abreviatura para FD es SIGLA	2	-	No
19	SIGLA (del inglés "FD")	2	-	No
20	SIGLA (del inglés FD)	2	-	No
21	SIGLA (siglas en inglés de FD)	2	-	No
22	SIGLA o FD	2	1	Si
23	SIGLA, que es la abreviatura de "FD"	2	-	Si
24	"FD", SIGLA	1	-	No
25	(FD, o SIGLA)	1	-	No
26	(FD/SIGLA)	1	-	No
27	(FD: SIGLA)	1	1	No
28	(SIGLA del inglés FD)	1	-	No
29	(SIGLA, "FD")	1	-	No
30	(SIGLA, de "FD")	1	-	No
31	(SIGLA, por "FD")	1	-	No
32	(SIGLA: FD)	1	1	No
33	FD ("SIGLA")	1	-	No
34	FD (abreviado como SIGLA)	1	-	No
35	FD (abreviado SIGLA)	1	-	No
36	FD, comúnmente conocido por SIGLA	1	-	No
37	FD, o de forma abreviada, SIGLA	1	-	No
38	la abreviatura SIGLA significa FD	1	-	No
39	SIGLA "FD"	1	-	No
40	SIGLA (abreviatura de FD)	1	-	No
41	SIGLA (acrónimo del inglés FD)	1	-	No
42	SIGLA (del inglés, "FD")	1	-	No

43	SIGLA (que abrevia FD)	1	-	No
44	SIGLA significa FD	1	-	Sí
45	SIGLA, abreviatura para "FD"	1	-	No
46	SIGLA, acrónimo de FD	1	-	No
47	SIGLA, de las iniciales inglesas de FD	1	-	No
48	SIGLA, FD.	1	-	Sí
49	SIGLA; FD	1	-	No

Tabla 11. Patrones de identificación de pares sigla-FD para el español

Algunos de estos patrones coinciden con los de los autores analizados más arriba; otros sólo se han hallado en nuestro corpus, lo cual puede deberse a las características propias del discurso de GH y MA en lengua española. Se destaca igualmente la amplia variedad de patrones empleados en los textos sobre Genoma humano frente a los empleados en los textos de Medio ambiente.

Tanto las reglas de formación como los patrones para la identificación de siglas son elementos esenciales a la hora de diseñar un extractor de siglas. Adicionalmente, cabe tomar decisiones sobre el método que se empleará para la adquisición. Como se ha mencionado antes, un sistema de detección-extracción puede basarse en: patrones, estadística, aprendizaje máquina o en una combinación de estos.

Independientemente del método que se emplee, el sistema deberá establecer la longitud de la ventana (contexto) a la derecha e izquierda del candidato a sigla, de modo que pueda identificarse el candidato a par sigla-FD. En la mayoría de los estudios se ha adoptado como longitud de ventana la siguiente: (|A|+5) o (|A|*2) palabras; aunque otros como Larkey *et al.* han establecido como criterio las 20 palabras anteriores y posteriores a la sigla.

Conclusiones

A lo largo de este capítulo se ha mostrado el estado de la cuestión en lo referente a los sistemas de detección y extracción de siglas. Del análisis realizado se derivan las siguientes conclusiones:

1. Las siglas son un fenómeno presente en todas las lenguas escritas, de ahí su carácter universal.
2. La motivación para investigar sobre técnicas y métodos de detección y extracción de este tipo de unidades proviene de las necesidades de ámbitos como: inteligencia artificial (IA), minería de datos (DM), recuperación de información (RI), y procesamiento del lenguaje natural (PLN).
3. El campo de la biomedicina es el que más esfuerzos ha dedicado a la investigación para el desarrollo de sistemas de extracción de siglas. Autores como Chang, Nadeau & Turney, Schwartz & Hearst, Adar, Dannélls y Zahariev, han trabajado en esta área. Otros ámbitos en los que se ha experimentado con extracción de siglas son: automoción, farmacéutica (Park & Byrd), biología molecular (Nenadić), prensa (Young) y medio ambiente (Taghva & Gilbreth).
4. Casi todos los sistemas de extracción de siglas estudiados se han creado para aplicarse a la lengua inglesa. A excepción del estudio de Dannélls (2005; 2006), orientado a las siglas de textos médicos en sueco, y del estudio de Zahariev (2004), la revisión de la bibliografía no ha arrojado luz sobre la existencia de sistemas similares para otras lenguas.
5. En la literatura revisada sobresalen dos motivaciones principales a la hora de crear un extractor de siglas: 1) para alimentar automáticamente BD de siglas y, de esta forma, mantenerlas actualizadas; y 2) para facilitar la tarea de extracción o recuperación de información en un campo de conocimiento dado.
6. El proceso de extracción de siglas consta de tres fases principales: 1) identificación de las siglas; 2) identificación de los pares sigla-FD, y 3) desambiguación.
7. Los sistemas de detección y extracción de siglas actuales se basan en tres métodos diferentes, a saber: 1) patrones; 2) estadística y 3) aprendizaje máquina. Aunque también se da el caso de sistemas híbridos.

8. Los sistemas basados en aprendizaje máquina junto con los híbridos se perfilan como los de mejor rendimiento. Sin embargo, no se debe pasar por alto que todos estos sistemas han sido pensados para analizar textos en lengua inglesa, por lo que se desconoce la eficacia de su aplicación en lenguas como el español.
9. A raíz de la circunstancia antes mencionada, se debe tener en cuenta que, por ejemplo, para el desarrollo de los patrones de detección de siglas en español, es necesario considerar un conjunto de patrones mixto, porque se pueden dar los siguientes casos:

    - la sigla y la FD aparecen en español
    - la sigla aparece en lengua extranjera y la FD en español
    - la sigla aparece en español y la FD en lengua extranjera

10. Los patrones más productivos para la identificación sigla-FD son:

    - FD (SIGLA)
    - SIGLA (FD)

11. Por último, un sistema para el español debe incorporar un número mayor de patrones de identificación de pares de sigla-FD, tal y como se indica en la tabla 11.

Bibliografía

Abbreviations.com [en línea]. (s.l.). http://www.abbreviations.com/about.asp [consulta: 6 de marzo de 2007].

Acronym Finder [en línea]. (s.l.). http://www.acronymfinder.com/ [consulta: 2 de octubre de 2006].

Acronym Server [en línea]. (s.l.). http://silmaril.ie/cgi-bin/uncgi/acronyms [consulta: 2 de octubre de 2006].

Acronyma [en línea]. (s.l.). http://www.acronyma.com/ [consulta: 2 de octubre de 2006].

Adar, E. (2004). «SaRAD: A simple and Robust Abbreviation Dictionary». En *Bioinformatics*. vol. 20, No. 4, 527-533- [En línea]. http://www.hpl.hp.com/research/idl/papers/srad/s-rad-090502.pdf [consulta: 4 de septiembre de 2004].

Akira, T.; Tokunaga, T. (2001). *Automatic disabbreviation by using context information*. [En línea]. *http://www.afnlp.org/nlprs2001/WS-Paraphrase/pdf/03-terada.pdf* [Consulta: 16 de marzo de 2006].

Ananiadou, S. *et al.* (2002). *Term-based Literature Mining from Biomedical Texts.* [En línea]. *http://www.pdg.cnb.uam.es/BioLink/Ananiadou.doc* [consulta 4 de septiembre de 2004].

Ao, H. (2005). *ALICE: An Algorithm to Extract Abbreviations from MEDLINE*. [En línea].*http://www.pubmedcentral.nih.gov/picrender.fcgi?artid=1205607&blobtype=pdf* [Consulta: 27 de octubre de 2006].

Benbasat, I.; Wand, Y. (1984). *Command Abbreviation Behavior in Human-Computer Interaction.* [En línea]. *http://portal.acm.org/citation.cfm?id=358027.358050* [Consulta: 5 de octubre de 2004].

Bracewell, D. *et al*. (2005). *Identification, Expansion, and Disambiguation of Acronyms in Biomedical Texts*. [En línea]. *https://troia.upf.edu/http/www.springerlink.com/content/p34858l370315765/fulltext.pdf* [Consulta: 2 de febrero de 2006].

Chang, J. *et al.* (2002). *Creating an Online Dictionary of Abbreviations from MEDLINE.* [En línea]. *http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pubmed&pubmedid=12386112* [Consulta: 22 de julio de 2004].

Dannélls, D. (2006). «Automatic Acronym Recognition». En *Proceedings of the 11th conference on European chapter of the Association for Computational Linguistics*. [En línea]. *http://www.cling.gu.se/~cl2ddoyt/pub/automatic.pdf* [Consulta: 16 de marzo de 2006].

_____, (2006). *Acronym Recognition: Recognizing acronyms in Swedish texts*. Tesis de máster. Göteborg: Department of Linguistics, Göteborg University. [En línea]. *http://www.cling.gu.se/~cl2ddoyt/pub/masterThes.pdf* [Consulta: 22 de octubre de 2006].

_____, (2005). *Recognizing Swedish acronyms and their definitions in biomedical literature*. [En línea]. *http://www.cling.gu.se/~cl2ddoyt/acronyms/report.pdf* [Consulta: 4 de julio de 2006].

_____, (2005). *Classifying Swedish Acronyms with MBT*. [En línea]. http://www.cling.gu.se/~cl2ddoyt/pub/mbl_project.pdf [Consulta: 4 de julio de 2006].

Dominich, S. *et al*. (2003). *A Study of the Usefulness of Institutions' Acronyms as Web Queries*. [En línea]. https://troia.upf.edu/http/www.springerlink.com/content/09dhtwfntgh5lq9k/fulltext.pdf [Consulta: 19 de diciembre de 2005].

Gaudan, S. *et al*. (2005). «Resolving abbreviations to their senses in Medline». En *Bioinformatics*. vol 21, No. 18, 3658-3664.

Hacohen-Kerner, Y. *et al*. (2004). *Baseline Methods for Automatic Disambiguation of Abbreviations in Jewish Law Documents*. [En línea]. https://troia.upf.edu/http/www.springerlink.com/content/y9ev3m8crhd7q12d/fulltext.pdf [Consulta: 2 de febrero de 2006].

Hahn, U. *et al*. (2005). *Cross-Language Mining for Acronyms and their Completions from the Web*. [En línea]. http://www.coling.uni-freiburg.de/~marko/publications/hahn-ds2005.pdf [Consulta: 16 de marzo de 2006].

Joshi, M. *et al*. (2006). *A Comparative Study of Supervised Learning as Applied to Acronym Expansion in Clinical Reports*. [En línea]. http://wsdgate.sourceforge.net/pubs/AMIA06JoshiM.pdf [Consulta: diciembre 5 de 2006].

Kan, M. (2003). *Metadata Extraction and Text Categorization Using Universal Resource Locator Expansions.* [En línea]. http://www.comp.nus.edu.sg/~kanmy/papers/tr103.pdf [Consulta: 4 de octubre de 2004].

Kiss, T.; Strunk, J. (2002). *Scaled log likelihood ratios for the detection of abbreviations in text corpora*. [En línea]. http://www.linguistics.rub.de/~kiss/publications/abbrev.pdf [Consulta: 16 de marzo de 2006].

Larkey, L. *et al.* (2000). *Acrophile: An Automated Acronym Extractor and Server.* [En línea].http://delivery.acm.org/10.1145/340000/336664/p205-larkey.pdf?key1=336664&key2=7455896901&coll=GUIDE&dl=GUIDE&CFID=28595879&CFTOKEN=50021223 [Consulta: 29 de marzo de 2003].

Liu, H. *et al*. (2002). *A Study of Abbreviations in MEDLINE Abstracts*. [En línea]. http://lhncbc.nlm.nih.gov/lhc/docs/published/2002/pub2002051.pdf [Consulta: 16 de marzo de 2006].

_____, (2001). *A Study of Abbreviations in the UMLS*. [En línea]. http://adams.mgh.harvard.edu/PDF_Repository/D010001239.pdf [consulta: 29 de noviembre de 2006].

Mima, H. *et al*. (2002). *A Methodology for Terminology-based Knowledge Acquisition and Integration*. [En línea]. http://acl.ldc.upenn.edu/coling2002/proceedings/data/area-16/co-228.pdf [Consulta: 5 de octubre de 2004].

Nadeau, D.; Turney, P. (2005). *A Supervised Learning Approach to Acronym Identification*. [En línea]. http://iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-48121.pdf [consulta: 1 de diciembre de 2006].

Narayanaswamy, M. *et al*. (2003) *A Biological Named Entity Recognizer.* [En línea]. *http://www.ccs.neu.edu/home/futrelle/bionlp/psb2003/narayanaswamy.pdf* [Consulta: 3 de marzo de 2004].

Nenadić, G. *et al.* (2006). «Towards a terminological resource for biomedical text mining». Actas del *International Conference on Language, Resources and Evaluation, LREC*, Génova, Italia. 1.071-1.076.

_____, (2003). *Terminology-driven Mining of Biomedical Literature.* [En línea]. *http://bioinformatics.oupjournals.org/cgi/reprint/19/8/938* [Consulta: 8 de mayo de 2004].

_____, (2002). «Automatic Acronym Acquisition and Term Variation Management within Domain-Specific Texts». Actas del *3rd International Conference on Language, Resources and Evaluation, LREC-3*, Las Palmas. 2.155-2.162.

_____, (2002). *Automatic Discovery of Term Similarities Using Pattern Mining.* [En línea]. *http://acl.ldc.upenn.edu/coling2002/workshops/data/w05/w05-08.pdf* [Consulta: 12 de marzo de 2004].

Okazaki, N.; Ananiadou, S. (2006). «Term Recognition Approach to Acronym Recognition. *Proceedings of the COLING/ACL*». [En línea]. *http://www.chokkan.org/publication/okazaki_COLACL2006.pdf* [Consulta: 28 de noviembre de 2006].

_____, (2006). «Building an Abbreviation Dictionary Using a Term Recognition Approach». En *Bioinformatics*. [En línea]. *http://bioinformatics.oxfordjournals.org/cgi/reprint/btl534v1* [Consulta: 28 de noviembre de 2006].

_____, (2006). «Clustering acronyms in biomedical text for disambiguation». [En línea]. *http://hnk.ffzg.hr/bibl/lrec2006/pdf/351_pdf.pdf* [Consulta: 30 de mayo de 2006].

Pakhomov, S. (2002). «Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts». En *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics ACL.* [En línea]. *http://nlp.cs.nyu.edu/nycnlp/P02-1021.pdf* [Consulta: 21 de enero de 2004].

Park, Y.; Byrd, R. (2001). *Hybrid Text Mining for finding Abbreviations and their Definitions.* [En línea]. http://www.research.ibm.com/talent/documents/emnlp2001_48.pdf [Consulta: 27 de octubre de 2004].

Pustejovsky, J. *et al.* (2001). *Linguistic Knowledge Extraction from Medline: Automatic Construction of an Acronym Database.* [En línea]. *http://www.medstract.org/papers/bioinformatics.pdf* [Consulta: 24 de mayo de 2004].

Schwartz, A.; Hearst, M. (2003). *A Simple Algorithm for identifying Abbreviation Definitions in Biomedical Text.* [En línea]. *http://biotext.berkeley.edu/papers/psb03.pdf* [Consulta: 26 de febrero de 2004].

Taghva, K.; Gilbreth, J. (1999). «Recognizing acronyms and their definitions». *International Journal on Document Analysis and Recognition*. 191-198. [En línea].*https://troia.upf.edu/http/www.springerlink.com/content/u6c9ymd1v8jflerh/fulltext.pdf* [Consulta: 30 de noviembre de 2006].

Torres, E.; Schulz, K. (2005). «Stable methods for recognizing acronym-expansion pairs: from rule sets to hidden Markov models». En *International Journal of Document Analysis.* Vol. 8, No. 1. 1-14.

Tsuruoka, Y.; Tsujii, J. (2003). *Probabilistic Term Variant Generator for Biomedical Terms.* [En línea]. http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/papers/sigir03.pdf [Consulta: 26 de febrero de 2004].

Tsuruoka, Y. *et al*. (2005). «A Machine Learning Approach to Acronym Generation». *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*. 25–31. [En línea]. http://acl.ldc.upenn.edu/W/W05/W05-1304.pdf [consulta: 30 de noviembre de 2006].

Valderrábanos, A. *et al.* (2002). *Textractor: a Multilingual Terminology Extraction Tool.* [En línea]. http://liquid.sema.es/document_pdf/textractor_a_multilingual_terminology_extraction_tool.pdf [Consulta: 26 de febrero de 2004].

Wiley Interscience [en línea]. (s.l.). http://www3.interscience.wiley.com/cgi-bin/home?CRETRY=1&SRETRY=0 [consulta: 2 de octubre de 2006].

Wren, J.; Garner, H. (2002). «Heuristics for Identification of Acronym-Definition Patterns Within Text: Towards an Automated Construction of Comprehensive Acronym-Definition Dictionaries». En *Methods of Information in Medicine*. Vol. 41, No. 5. 426-34. [En línea]. http://www.schattauer.de/index.php?id=739&no_cache=1&artikel=413 [Consulta: 12 de febrero de 2007].

Xu, J.; Huang, Y. (2006). «Using SVM to extract acronyms from text». En *Soft Comput*. 11. 369-373. [En línea]. http://www.springerlink.com/content/276028782h150080/fulltext.pdf [Consulta: 12 de febrero de 2007].

_____, (2005). «A Machine Learning Approach to Recognizing Acronyms and their Expansion». En *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*. [En línea]. http://nkxujun.googlepages.com/AcronymExtraction-ICMLC2005.pdf [consulta: diciembre 1 de 2006].

Yeates, S. (1999). *Automatic Extraction of Acronym from Text.* [En línea]. http://www.cs.waikato.ac.nz/~nzdl/publications/1999/yeates-Auto-Extract.pdf [Consulta: 5 de julio de 2004].

Yeates, S. *et al.* (2000). *Using Compression to identify Acronyms in Text.* [En línea]. http://arxiv.org/PS_cache/cs/pdf/0007/0007003.pdf [Consulta: 20 de marzo de 2004].

Yi, J.; Sundaresan, N. (1999). *Mining the Web for Acronyms Using the Duality of Patterns and Relations.* [En línea]. https://troia.upf.edu/http/delivery.acm.org/10.1145/320000/319782/p48-yi.pdf?key1=319782&key2=1164994611&coll=portal&dl=ACM&CFID=7731795&CFTOKEN=76752898 [Consulta: 16 de febrero de 2004].

Yoshida, M. *et al*. (2000). «PNAD-CSS: A workbench for constructing a protein name abbreviations dictionary». En *Bioinformatics*. vol 16, Nº 2. 169-175. [En línea]. http://bioinformatics.oxfordjournals.org/cgi/reprint/16/2/169 [Consulta: 27 de octubre de 2006].

Young, A. (2004). *Automatic Acronym Identification and the Creation of an Acronym Database.* [En línea]. http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/ug2004/pdf/u1ay.pdf [Consulta: 31 de marzo de 2005].

Yu, H. *et al.* (2006). *A Large Scale, Corpus-Based Approach for Automatically Disambiguating Biomedical Abbreviations*. [En línea]. http://delivery.acm.org/10.1145/1170000/1165778/p380-yu.pdf?key1=1165778&key2=5451994611&coll=ACM&dl=ACM&CFID=15151515&CFTOKEN=6184618 [Consulta: 7 de noviembre de 2006].

_____, (2002). *Mapping Abbreviations to Full Forms in Biomedical Articles.* [En línea]. http://www1.cs.columbia.edu/~hongyu/paper/JAMIA_02_Mapping.pdf [Consulta: 2 de abril de 2004].

Yu, H.; Agichtein, E. (2003). «Extracting Synonimous Gene and Protein Terms from Biological Literature». *Bioinformatics*. vol 1, Nº 1. 1-10.

Yu, Z. *et al.* (2003). *Automatic Resolution of Ambiguous Abbreviations in Biomedical Texts using Support Vector Machines and One Sense Per Discourse Hypotesis.* [En línea]. http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/papers/sigir03bio.pdf [Consulta: septiembre 18 de 2004].

Zahariev, M. (2004). *A(Acronyms)*. Tesis doctoral. Burnaby: School of Computing Science, Simon Fraser University. [En línea]. http://www.cs.sfu.ca/~manuelz/personal/p/f.pdf [Consulta: octubre 5 de 2004].

_____, (2004). «A Linguistic Approach to Extracting Acronym Expansions from Text». En *Knowledge and Information Systems*. 6. 366-373. [En línea] http://www.springerlink.com/content/mt4q8d4rhk8f667r/fulltext.pdf [Consulta: 10 de febrero de 2006].

_____, (2003) *Efficient Acronym-Expansion Matching for Automatic Acronym Acquisition.* [En línea]. http://www.cs.sfu.ca/~manuelz/personal/p/da.pdf [Consulta: octubre 5 de 2004].