Quid, N°. 22, pp. 23-30, Ene-Jun, 2014, ISSN: 1692-343X, Medellín-Colombia

USO DE ONTOLOGÍAS PARA LA DESAMBIGUACIÓN DE ARTEFACTOS DE SOFTWARE EN LA EDUCCIÓN DE REQUISITOS DE SOFTWARE

USING ONTOLOGIES FOR DISAMBIGUATION OF SOFTWARE ARTIFACTS IN SOFTWARE REQUIREMENTS ELICITATION

Sebastián Alonso Gómez Arias
Universidad Nacional De Colombia,
Facultad de Minas,
Maestría en Ingeniería de Sistemas.
SintelWeb B1 M8A Oficina 306,
Medellín, Colombia, seagomezar@unal.edu.co

Jaime Alberto Guzmán-Luna
Universidad Nacional De Colombia,
Facultad de Minas, Doctor en Ingeniería.
SintelWeb Bl M8A Oficina 306,
Medellín, Colombia,
jaguzman@unal.edu.co

(Recibido el 12-01-2014. Aprobado el 20-03-2014)

Resumen: el problema de ambigüedad semántica es transversal a diversas áreas del conocimiento, entre ellas la ingeniería de requisitos. En muchos casos los requisitos de un software se encuentran plagados de ambigüedad debido a la vaguedad misma del lenguaje. En este artículo se presentan algunas técnicas de desambiguaci6n semántica, entre ellas algunas técnicas basadas en ontologías, y se argumenta por que las técnicas basadas en ontologías deben ser usadas en el proceso de educción de requisitos de software, y se presenta un caso práctico donde se hace una desambiguación de una especificación de requisitos usando una técnica basada en conocimiento: el algoritmo Lesk.

Palabras clave: ingeniería de Requisitos, Ambigüedad Semántica, Desambiguación Semántica, Ontología, Algoritmo Lesk.

Abstract: the problem of semantic ambiguity is transverse to various areas of knowledge, including requirements engineering. In many cases the requirements of a software are riddled with ambiguity due to the vagueness of the language, in this paper some technical semantic disambiguation are presented, including some techniques based on ontologies, and argues why the techniques based on ontologies should be used in the process of software requirements elicitation, and a practical case where a disambiguation of a requirements specification using a technique based on knowledge is presented: the algorithm Lesk.

Keywords: requirements engineer, Semantic Ambiguity, Word Sense Disambiguation, Ontology, Lesk Algorithm.

1. INTRODUCCION

La educción de requisitos de software es una actividad de la primera fase del ciclo de vida del software. Su importancia está dada porque permite que los usuarios y los analistas identifiquen los requisitos que debe cumplir un software. Los problemas en el proceso de educción de requisitos son: (i) la comunicación limitada o insuficiente entre el interesado (usuario) y el analista, generada por las diferencias entre sus especialidades; (ii) la ambigüedad verbal durante el discurso; y (iii) el analista es quien elabora subjetivamente los requisitos del software a partir de la información suministrada por el interesado. Véase Figura 1.

Fig 1. Proceso natural de educción de requisitos

El analista debe realizar la identificación de requisitos y validarlos con el interesado una vez comprenda la naturaleza, características y límites del problema o la necesidad. Sin embargo, las diferencias que existen entre las especialidades de estos, hace que dicha validación no sea interpretada de la misma manera entre las partes (ambigüedad), lo que finalmente se traduce en la entrega de una pieza de software que, en la mayoría de los casos, no cumple con las necesidades que quiso plantear el interesado. Véase Figura 2.

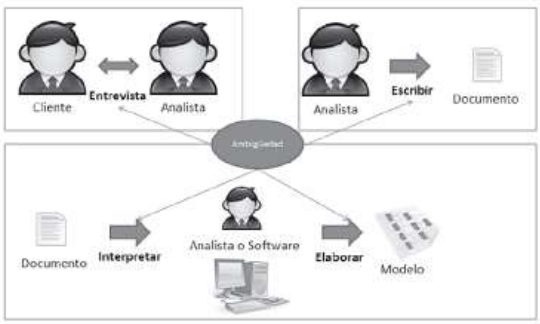

Fig. 2. Problema de ambigüedad en la educci6n de requisitos

Este trabajo busca proponer un mecanismo para enfrentar la ambigüedad verbal dada en un contexto usando el algoritmo Lesk, con el fin de minimizar errores en la identificación de requisitos de software por esta ambigüedad verbal.

2. TECNICAS DE DESAMBIGUACION

Los métodos de desambiguación de sentidos de las palabras se pueden clasificar en tres tipos según.

I. Supervisados
II. No supervisados
III. Basados en Conocimiento

Supervisados: estos métodos de desambiguaci6n utilizan técnicas de aprendizaje de maquina con el fin de aprender e inferir reglas a partir de corpus ya entrenados (corpus significan textos etiquetados y desambiguados previamente) con el fin de, con base en las reglas inferidas, procesar y desambiguar textos nuevos. En la tabla 1 se enuncian algunos de estos métodos.

Tabla 1. Algunos métodos de desambiguación supervisados

Método de desambiguación	Autor(es)
Listas de decision	(Rivest, 1987), (Yarowsky, 2000)
Arboles de decision	(Quinlan, 1986)
Redes Bayesianas	(Escudero, Márquez, Rigau, & Salgado, 2000)
Redes neurales	(Tsatsaronis, Vazirgiannis, & Androutsopoulos, 2007)
Basados en ejemplos Basados en memoria	(Ng, 1997), (Daelemans, Van Den Bosch, & Zavrel, 1999)
Maquinas vectoriales	(Boser, Guyon, & Vapnik, 1992)
Métodos de ensamble	(Florian, Cucerzan , Schafer, & Yarowsky , 2002)

No Supervisados: Estos métodos de desambiguación utilizan tan solo el contexto de una palabra con el fin de desambiguar la palabra en cuesti6n, se basan en la idea de que el mismo sentido de una palabra tendrá palabras vecinas similares.

Basados en conocimiento: Estos métodos pretenden explotar el conocimiento (tesauros, ontologías, diccionarios etc.) con el fin de inferir el sentido de una palabra en un contexto.

Tabla 2. Algunos métodos de desambiguación no supervisados

Método de desambiguación	Autor(es)
Agrupamiento de contexto	(Schutze, 1992)
Agrupamiento de palabra	(Lin, 1998)
Grafos de concurrencia	(Widdows & Dorow, 2002)

Estos métodos hacen uso de WordNet, la cual es una base de datos léxica que agrupa palabras en conjuntos de sinónimos llamados synsets, proporcionando definiciones cortas y generales, y almacena las relaciones semánticas entre los conjuntos de sinónimos. Su prop6sito es doble: producir una combinación de diccionario y tesauro cuyo uso sea más intuitivo y soportar análisis automático de texto y a aplicaciones de Inteligencia Artificial.

La relación de hiperonimia e hiponimia entre los sentidos de cada palabra puede ser interpretada como una relación de especialización entre categorías conceptuales. En otras palabras, WordNet puede ser interpretado y usado como una ontología en las Ciencias de la Computación y para las tareas de desambiguaci6n -(Snasel, Moravec, & Pokorny, 2005).

Dado que el enfoque de este trabajo está basado en ontologías, a continuación presentamos algunos métodos basados en ontologías.

Método de desambiguación	Autor(es)
Simplified Lesk	(Lesk, 1986)
MFS	(Preiss, Dehdari, King, & Mehay, 2009)
Distancia Conceptual	(Lewis, 2001)

3. DESAMBIGUACION BASADA EN CONOCIMIENTO

3.1. Lesk

El algoritmo Lesk se presentó por primera vez en 1986, este tiene como propósito principal determinar el mejor synset posible para una palabra en una oración dada -(Lesk, 1986).

Este algoritmo se basa en la suposición de que las palabras de un vecindario de una porción de texto tenderán a compartir un tema en común y, por tanto, es posible identificar el sentido más correcto con base en las definiciones del diccionario.

Una version simplificada del algoritmo de Lesk, es comparar la definición de Wordnet de una palabra ambigua con los demás términos contenidos en su vecindario. Su representaci6n formal se presenta a continuación.

$$ScoreLesk(S)\ context\ (w)\ 1Voss(S) \quad (1)$$

Donde context (w) representa el conjunto de palabras de la frase donde está contenida la palabra a desambiguar.

El algoritmo Lesk debe cumplir con las siguientes reglas básicas: (i) para todos los sentidos de la palabra que se desea desambiguar, se debe contar la cantidad de palabras que se encuentran en el vecindario de la palabra ambigua y en la definición de cada sentido de la palabra en el diccionario seleccionado, (ii) el sentido que ha de ser elegido, será aquel que tenga el mayor número en el conteo realizado, y (iii) no participan en el análisis las palabras vacías o "stop words" que son aquellas palabras diferentes a verbos, adjetivos o sustantivos.

El siguiente es el seudocódigo del algoritmo utilizado para desambiguar palabras en una frase:

```
Functiondesambiguar (frase) returnsset_of_senses
                    i=0;
                    foreach palabra in frase

                    sentidopalabra=
Lesk(palabra,frase);

                    set_of_senses[i]=sentidopalabr
a;
                                        i=i+1;
                    endforeach
                    returnset_of_senses;
endfunction

Function1.esk (palabra, frase) returnsmejor_sentido
                    mejor_sentido =
sentido_mas_frecuente;
                    contador_maximo = 0;
                    contexto =
conjunto_de_palabras_en_la_frase;
                    foreachsentido in sentidos of
palabra

                    signature =
conjunto_de_palabras_en_la_definicion_del_sentido;

                    contador =
contar(signature,context);
                                        if
(contador>contador_maximo) then

                    contador_maximo=contador;

                    mejor_sentido=sentido;
                                        endif
                    endforeach
                    returnmejor_sentido;
endfunction
```

3.2. Algoritmo MFS (Most Frequent Sense)

El sentido más frecuente (o MFS, por sus siglas en ingles), es un algoritmo de desambiguación semántica polisémica, que permite encontrar el sentido mayormente utilizado en un corpus, para cada palabra.

Este método parte de un corpus anotado con sentidos, del cual se aprenden las frecuencias de sentidos individuales. Para cada palabra a desambiguar, un etiquetador morfosintáctico se usa para determinar la etiqueta PoS (Part-of-Speech), y luego se selecciona el sentido para esa etiqueta. Aunque es un método sencillo, ha mostrado ser difícil de superar, debido a que presenta un desempeño de 62,5%. De hecho, tan solo 5 de los 26 sistemas enviados al *Senseval3* han logrado mejores resultados (Preiss et al., 2009).

El éxito del MFS se debe principalmente a la distribución de la frecuencia de los sentidos, en función de la índice del sentido versus un grafo de frecuencias. En este grafo incluso obtiene una curva tipo Zipf en la que los sentidos mejor calificados tienen mayor probabilidad de ser el sentido correcto (Preiss et al., 2009).

Sin embargo, como reportan (Snyder & Palmer, 2004), los resultados dependen, entre otros factores, de la efectividad del etiquetador morfosintáctico. Lo demuestran con los siguientes resultados experimentales:

Si las palabras poseen el lema y el etiquetado PoS correctos, el MFS arroja un desempeño del 66%.

En caso de que las palabras utilicen algún algoritmo de tematización, y ninguna información del etiquetado PoS, se puede alcanzar un desempeño de 32%.

Sin embargo, en la literatura se han desarrollado algunas modificaciones con el fin de mitigar estas desventajas. Por ejemplo, Helmut Schmid (Schmid, 1994), propone una estrategia basada en los lemas, que permite mejorar el desempeño hasta un 56%.

Por su parte, McCarthy, Koeling y Carroll (McCarthy, Koeling, & Carroll, 2007), muestran que es posible obtener el sentido predominante, es decir, el más frecuente, para una palabra en un corpus sin necesidad de tener los sentidos etiquetados en el corpus. Proponen utilizar el tesauro creado automáticamente en (Lin, 1998), y la siguiente métrica para asignar una puntuación a cada sentido *s*, de cada palabra w:

$$\sum_{n_j \in N_w} dss(w, n_j) * \frac{sss(s_i, n_j)}{\sum_{s_i' \in \, senses\,(w)} sss(s_i', n_j)}$$

donde *dss (w, n)* representa la distribución de la similitud de una palabra w en la vecindad del tesauro w; la función *sss* ($s_p n_i$) define la máxima similitud entre el sentido *s*, de la palabra w, y un sentido s_x del vecindario n_i de la misma palabra en el tesauro. Esta función se define formalmente con la siguiente ecuación:

$$sss(s_i, n_j) = max_{s_x \in \, senses\,(w)} sss(s_i', n_j)$$

McCarthy et al. sostiene que este método supera a la propuesta original del MFS, solo en el caso de tener una frecuencia de la palabra en el corpus menor a 5. En (Preiss et al., 2009) se propone seguirla solo bajo esa condición, y si no se cumple, se utiliza el coeficiente de Bhattacharyya.

3.3. Distancia Conceptual

La distancia conceptual pretende arrojar una base para medir la cercanía en el significado entre palabras, tomando como referencia una red jerárquica estructurada (Gónzalez-Agirre, Laparra, & Rigau, 2012).

Para ello, la principal representaci6n de palabras está dada por WordNet. Como se ha explicado anteriormente, WordNet es una ontología que representa alrededor de 70,000 entradas léxicas del inglés, y su estructura es de una red semántica con términos enlazados a través de relaciones de antónimos, sinónimos, homónimos, meronimos, hiperonimos, e hiponimos. Sin embargo, para la distancia conceptual, son importantes aquellos que establezcan relaciones tipo IS-A (es-un), los cuales son las relaciones de hiperonimos e hiponimos (Lewis, 2001).

Ahora bien, para medir una distancia conceptual en dicha red, aparecen propuestas como la que se presenta en (Rada, Mili, Bicknell, & Blettner, 1989). Allí, se presenta una descripción de una métrica usada para medir la distancia conceptual en una red semántica. Allí, la distancia se calcula midiendo la distancia entre nodos (cada nodo representa un concepto), contando el flamero de nodos que hay en el camino. Así, los nodos que tienen una menor distancia, son más cercanos semánticamente.

Este algoritmo propuesto por Rada et al., es bastante simple. Basta con recorrer el espacio semántico entre dos nodos, empezando en uno, y terminando cuando se llegue al otro. Sin embargo, esta estrategia no necesariamente arroja el camino más corto. Por eso, Lewis (Lewis, 2001) sugiere buscar el camino más corto, que resulta más adecuado para la medición de la distancia conceptual que simplemente cualquier ruta. De esta forma, en vez usar una estrategia

primero en profundidad *(depth-first),* se implementa una estrategia primero en amplitud *(breadth-first)* se convierte en un mejor enfoque, ya que simultáneamente se expanden todas las conexiones en cada rama, y se detiene cuando se encuentre el nodo buscado.

Se puede concluir que la distancia conceptual es una técnica usada para determinar qué tan cerca en la ontología de Wordnet se encuentran dos conceptos, esto se mide contando los conceptos que se encuentran en el camino más corto entre los dos nodos a comparar (Navigli, 2009).

Este cálculo de distancia conceptual fue adaptado por -(Aguirre & Rigau, 1996) para realizar tareas de desambiguación, debido a que en una frase con palabras polisémicas, es posible desambiguar completamente la frase identificando los sentidos que hacen parte del camino más corto que une todos los nodos de la oración.

4. EXPERIMENTOS

Para los experimentos se implementó el algoritmo Simplified Lesk en el lenguaje JAVA y se consumió la ontología de wordnet como fuente de los significados para el algoritmo. También se utilizaron los requisitos de software propuestos en el caso de estudio "Controlador Avanzado y Automático de trenes" propuesto por (Letier, 2001) con el fin de desambiguar algunas palabras ambiguas presentes en este caso de estudio.

"El Sistema automático de control tiene que garantizar el control y la aceleración de los trenes para mantener el buen desplazamiento de estos.

El Sistema Automático de Control garantiza al tren su velocidad controlada.

El Sistema Automático de Control garantiza al tren la aceleración controlada.

El Sistema Automático de Control garantiza al tren que la distancia entre trenes se cumpla durante todo el recorrido.

El Sistema Automático de Control garantiza al tren el envió de la señal para establecer si se permite o no el ingreso de un tren a un tramo de la vía. "

En la figura 3 se presentan estos requisitos y se subrayan las palabras ambiguas.

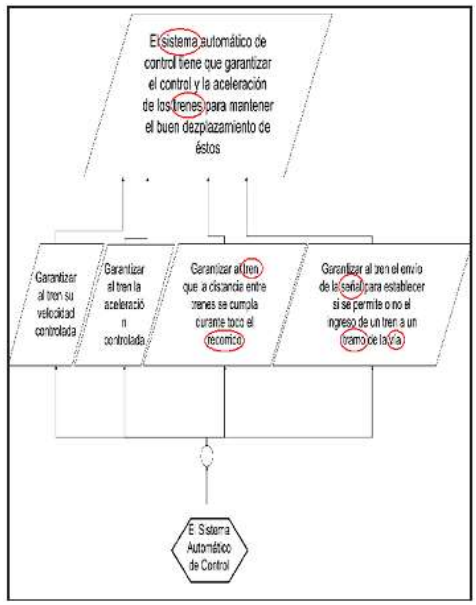

Fig. 3. Requisitos de software ambiguos en el caso de estudio "Controlador Avanzado de trenes"

En la tabla 3 se presentan las palabras desambiguadas utilizando el algoritmo Lesk presentado en este enfoque.

Tabla 3 Palabras desambiguadas con el algoritmo Lesk

Palabra	***Sentido***	***Significado***
Sistema	No Identificado	No Identificado
Tren	spa-30-04468005-n	public transport provided by a line of railway cars coupled together and drawn by a locomotive:
Recorrido	spa-30-01212230-v	pass over, across, or through:
Serial	spa-30-06791372-n	any nonverbal action or gesture that encodes a message
Tramo	spa-30-13939892-n	a specific identifiable position in a continuum or series or especially in a process
Vía	spa-30-04564698-n	any artifact consisting of a road or path affording passage from one place to another

Como se puede observar en la tabla 3 algunas palabras no pudieron ser desambiguadas como el caso de "Sistema".

5. CONCLUSIONES

En este artículo se presentaron los tres tipos de métodos de desambiguación de sentidos de la palabras (Supervisado, no supervisado y basado en conocimiento), pero se profundizo en los métodos basados en conocimiento debido a que estos hacen use de ontologías y, por ende, tienen un alcance y cobertura de los idiomas mucho más amplio que los demás métodos. Adicionalmente, no requieren de ningún entrenamiento previo, aunque su costo computacional sea un poco más elevado.

Dado lo anterior, se use el algoritmo Lesk con el fin de desambiguar requisitos de software del caso de estudio del controlador de trenes y se presentaron los resultados. Como trabajo futuro se pretende incrementar las técnicas de desambiguaci6n usadas y probar con muchos más casos de estudio con el fin de verificar la eficiencia de estos métodos de desambiguación basados en ontologías en el proceso de educción de requisitos de software.

6. RECONOCIMIENTO

Este artículo se realizó en el marco del proyecto de investigación: Modelo de procesamiento terminológico basado en ontologías para la desambiguación verbal en la educción de requisitos de software, Dirección de Investigaciones de la Universidad Nacional de Colombia Sede Medellín DIME, a través de la Convocatoria del Programa Nacional de Proyectos para el Fortalecimiento de la Investigación, la Creación y la Innovación en Posgrados de la Universidad Nacional de Colombia 2013-2015.

7. REFERENCIAS

Aguirre, E., & Rigau, G. (1996). Word sense disambiguation using conceptual density. En: *Proceedings of the 16th conference on Computational linguistics Volume 1* (pp. 16-22). Association for Computational Linguistics. Recuperado a partir de http://d1.acm.org/citation.cfm?id=992635

Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (1999). *El lenguaje unificado de modelado* (Vol. 1). Addison-Wesley. Recuperado a partir de http://pwp.etb.net.co/witorres/poo/3E-UML.pdf

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. En *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). ACM. Recuperado a partir de http://d1.acm.org/citation.cfm?id=130401

Daelemans, W., Van Den Bosch, A., & Zavrel, J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3), 11-41.

Escudero, G., Márquez, L., Rigau, G., & Salgado, J. G. (2000). On the portability and tuning of supervised word sense disambiguation systems. Recuperado a partir de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.3361

Florian, R., Cucerzan, S., Schafer, C., & Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8(4), 327-341.

González-Aguirre, A., Laparra, E., & Rigau, G. (2012). Multilingual Central Repository version 3.0. *Proceedings of the 8th International Conference on Language Resources and Evaluation*, 2525-2529.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. En *Proceedings of the 5th annual international conference on Systems documentation* (pp. 24-26). ACM. Recuperado a partir de http://dLacm.org/citation.cfm?id=318728 Letier, E. (2001). *Reasoning about agents in goal-oriented requirements engineering*. PhD thesis, Universite catholique de Louvain. Recuperado a partir de http://www0.cs.ucl.ac.uk/staff/e.letier/publications/letier-thesis.pdf

Lewis, W. D. (2001). Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring. *Language in Cognitive Science*, *12*, 9-16.

Lin, D. (1998). Automatic retrieval and clustering of similar words. En *Proceedings of the 17th international conference on Computational linguistics* (Vol. 2, pp. 768-774). Association for Computational Linguistics.

McCarthy, D., Koeling, R., & Carroll, J. (2007). Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4), 553-590.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR), 41(2),* 10.

Ng, H. T. (1997). Getting serious about word sense disambiguation. En *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How* (pp. 1-7). Recuperado a partir de http://aclidc.upenn.edu/W/W97/W97-0201.pdf

Preiss, J., Dehdari, J., King, J., & Mehay, D. (2009). Refining the most frequent sense baseline. En *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions* (pp. 10-18). Boulder, Colorado: Association for Computational Linguistics.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning, 1(1),* 81-106.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics, 19(1),* 17-30. doi:10.1109/21.24528

Rivest, R. L. (1987). Learning decision list. *Machine learning, 2(3),* 229-246.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. En *Proceedings of International Conference on New Methods in Language Processing* (pp. 44-49). Manchester, UK.

Schutze, H. (1992). Dimensions of meaning. En *Supercomputing'92., Proceedings* (pp. 787-796). IEEE. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=236684

Snasel, V., Moravec, P., & Pokorny, J. (2005). WordNet ontology based model for web retrieval. En *Web Information Retrieval and Integration, 2005. WIRI'05. Proceedings. International Workshop on Challenges in* (pp. 220-225). IEEE. Recuperado a partir de: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1553017

Snyder, B., & Palmer, M. (2004). The English all-words task. En T. Mihalcea, R. and Chklowski & Editors (Eds.), *Proceedings of SENSEVAL-3: Third In- ternational Workshop on Evaluating Word Sense Disambiguating Systems* (pp. 41-43). Barcelona, España.

Tratsaronis, G., Vazirgiannis, M., & Androutsopoulos, I. (2007). Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. En *IJCAI* (Vol. 7, pp. 1725-1730). Recuperado a partir de http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-279.pdf

Widdows, D., & Dorow, B. (2002). A graph model for unsupervised lexical acquisition. En *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1-7). Association for Computational Linguistics. Recuperado a partir de http://d1.acm.org/citation.cfm?id=1072342

Yarowsky, D. (2000). Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1-2), 179-186.