



## TRADUCTOR DE ESPAÑOL A METALENGUAJE SEMÁNTICO NATURAL

## CONVERTER FROM SPANISH TO NATURAL SEMANTIC METALANGUAGE

León Enrique Verdín-Armenta<sup>1</sup> y Miriam Díaz-Rodríguez<sup>2</sup>

<sup>1</sup>Egresado de Maestría en Sistemas Computacionales. Departamento de Investigación y Posgrados. Instituto Superior de Jalisco José Mario Molina Pasquel y Henríquez Unidad Académica Zapopan. <sup>2</sup>Profesora de tiempo completo Titular A. Departamento de Investigación y Posgrados. Instituto Superior de Jalisco José Mario Molina Pasquel y Henríquez Unidad Académica Zapopan.

## RESUMEN

Se desarrolla un programa capaz de convertir un subconjunto del español a vocabulario del Metalenguaje Semántico Natural. Este metalenguaje cuenta con las características de ser libre de ambigüedades, de estar formado por conceptos simples y tener un vocabulario reducido, lo cual presenta ventajas para su procesamiento por computadora. Para mostrar cómo esta información se puede usar y que una reducción en la complejidad del texto es ventajosa, se hace una pequeña implementación de un sistema experto que es capaz de aprender las reglas del juego de gato y de evaluar el tablero para determinar el resultado de la partida.

**Palabras clave:** Lenguaje natural, NSM, juego, instrucciones, ambigüedad

## SUMMARY

The aim of this project is to develop a program able to convert a subset of Spanish words to Natural Semantic Metalanguage vocabulary. This metalanguage is free of ambiguities, it is formed with simple concepts and has a reduced vocabulary, making it convenient for its processing by computer. To show how this information can be used and that a reduction in the complexity is advantageous, a small implementation of an expert system with the ability of learning the tic-tac-toe rules and evaluating the board to determine the game status is made.

**Key words:** Natural language, NSM, game rules, instructions, ambiguity.

## INTRODUCCIÓN

La comunicación entre las personas y los sistemas computacionales es cada vez más frecuente y directa, lo cual puede ser problemático debido a la gran diferencia en que la información es captada y procesada por los seres humanos y las máquinas. Muy pocos sistemas se basan en la interpretación directa de la información, esto por la dificultad de garantizar la calidad de la información de entrada.

Este problema pertenece al procesamiento de lenguaje natural (PLN), que es un sub-campo de la lingüística y la inteligencia artificial (IA) y estudia los problemas inherentes al procesamiento y manipulación del lenguaje natural. El objetivo final del PLN es hacer que las computadoras “entiendan” ideas expresadas en lenguaje humano (Linckels & Meinel, 2011). Una herramienta que usa ampliamente el PLN son los sistemas expertos, usados para mejorar la interacción entre los usuarios y la información que existe en una base de datos.

El desarrollo de estos sistemas es complejo y por lo general se prefiere hacerlos en inglés. Debido a lo anterior, el objetivo de este proyecto es demostrar que el español, convertido a un vocabulario reducido del mismo y sin ambigüedad puede ser usado por un sistema experto directamente, facilitando su proceso de desarrollo. Con el método usado para desarrollar este programa traductor de español a Metalenguaje

Semántico Natural (Natural Semantic Metalanguage o NSM), se obtiene como resultado un texto con atributos que facilitan su procesamiento como la falta de ambigüedad sintáctica y semántica, un vocabulario limitado a un conjunto pequeño de primitivas semánticas y gramática más simple que el texto original.

En la implementación del ejemplo se usa como texto de entrada la explicación de las reglas en español sencillo del juego de *gato* y el texto resultante (ya convertido a vocabulario NSM) se usa para alimentar un motor de inferencia de un sistema experto.

### *Sistemas expertos*

Los sistemas expertos son programas de computadora que resuelven problemas complejos dentro de un área específica. El motor de inferencia del sistema experto obtiene la información necesaria del usuario, corre los modelos si se necesitan, e interpreta la salida para el usuario. Los datos son los hechos o las evidencias con las que se va a trabajar en un problema. Es posible incrementar los datos iniciales a través de la incorporación de las nuevas conclusiones que producen las reglas, si es el caso. Algunas de las reglas de inferencia más utilizadas son Modus Ponens y Modus Tollens y se pueden construir reglas más complejas mediante el *encadenamiento de reglas*. Esta estrategia puede utilizarse cuando las premisas de ciertas reglas coinciden con las conclusiones de otras. Esto se repite sucesivamente hasta que no pueden obtenerse más conclusiones.

### *Redes semánticas*

John F. Sowa describe las redes semánticas como estructuras de grafo que representan el conocimiento en patrones de nodos y arcos interconectados (Sowa, 2015); una red semántica forma parte de la estructura de datos del sistema experto. En computación el término “redes semánticas” se usó por primera vez en 1956 por Richard Richens para la pre-programación de un sistema de traducción por computadora (Sowa, 2016), y posteriormente en 1968 por Ross Quillian en su tesis doctoral para hablar sobre la organización de la memoria semántica humana o memoria para conceptos de palabras (Sharples, 1989).

### *Primitivas semánticas*

Siendo las redes semánticas modelos relacionales, la estrategia usual de los diseñadores de redes semánticas es hacer que el número de primitivas relacionales sea mínimo (TEMA, AGENTE, SUJETO, ATRIBUTO), mientras que no hay límite para el número de tipos (como sustantivos, verbos e incluso oraciones completas) (Malrieu, 2002). Las primitivas semánticas atacan el problema de la circularidad en las definiciones. Este problema consiste en que, al usar los diccionarios comunes al explicar un término, pueden usarse otros términos que a su vez necesitan definirse, y eventualmente se llega a las palabras iniciales. Las primitivas semánticas evitan este problema ya que se tiene un conjunto de palabras o conceptos que son considerados como básicos, es decir, que ya no es posible definirlos con otras palabras. Algunos tipos de primitivas semánticas que se desarrollaron fueron la dependencia conceptual de Roger Schank, KL-ONE de Brachman et al, Web Ontology Language (OWL) del grupo de trabajo de OWL y una lista más lingüísticamente inspirada es desarrollada por Wierzbicka, el Metalenguaje Semántico Natural (NSM) (Kornai, 2010).

En (Di Eugenio, 1992), Barbara Di Eugenio usa las primitivas de Jackendoff propuestas en (Jackendoff, 1990) para desarrollar un modelo computacional que adapta el plan al objetivo, y viceversa y después lo usa para que el sistema tenga capacidad de seguir instrucciones expresadas como imperativos; Di Eugenio continúa con la integración de un sistema híbrido y de estructuras conceptuales, y dice: *He mostrado que dos esquemas de representación diferentes son útiles para realizar diferentes tipos de inferencia. [...] Dado*

*que los dos formalismos son promisorios, el siguiente paso natural es integrarlos, ambos se beneficiarán de esta integración.*

En (Goddard, *et al.*, 2006) se da el listado las primitivas semánticas de la teoría NSM en español, algunas son:

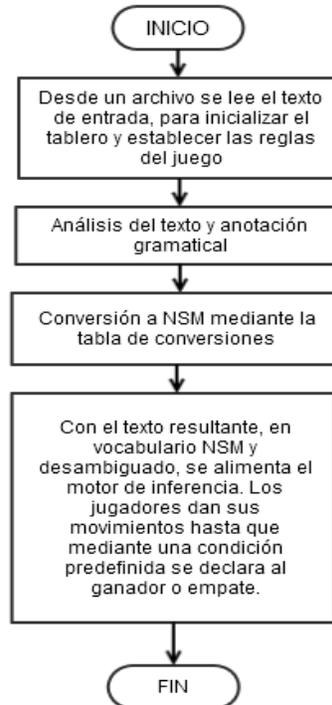
YO, TÚ, ALGUIEN~PERSONA, ALGO~COSAS, ESTO, LO MISMO, OTRO, UNO, DOS, ALGUNOS, MUCHO, POCO, TODO, BUENO, MALO, GRANDE, PEQUEÑO, SABER, OÍR, VER, HACER, MOVERSE, ESTAR (EN ALGÚN LUGAR), HAY, SER (ALGUIEN/ALGO), (ALGO) ES (DE ALGUIEN), DONDE~LUGAR, AQUÍ, ARRIBA, DEBAJO, LADO, DENTRO, NO, SI, TIPO, PARTE, COMO.

Kristin Stock et al en (Stock, et al., 2011), mediante el uso del metalenguaje semántico natural crea un sistema de consulta de información geográfica y usa varias palabras de ubicación geográfica que forman parte del vocabulario del NSM.

## MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN

Los pasos para la traducción de forma general se pueden ver en la *Figura 1*. En esta sección se detallan en las siguientes subsecciones las características de la entrada, el proceso de traducción y la conversión a NSM.

### *Lectura de reglas y anotación gramatical*



**Figura 1.** Algoritmo general de la investigación.

Se toman las instrucciones generales desde un archivo con texto como el del *Cuadro 1*. La interpretación de algunas reglas puede depender de la existencia de otras reglas, por lo que el orden de las instrucciones es importante. El procesamiento de estas reglas, y su integración final al sistema experto, es parecido en todos los casos de texto de entrada. El texto en este archivo se separa en oraciones y estas en palabras. Después es procesado por un módulo de Freeling para encontrar sus etiquetas morfosintácticas y esta información es recopilada en una estructura de datos.

Freeling es una librería de código abierto para el procesamiento multilingüe automático, que proporciona una amplia gama de servicios de análisis y anotación lingüísticos. Es personalizable y ampliable, y está orientado a aplicaciones del mundo real en términos de velocidad y robustez (Padro, 2012).

El etiquetado gramatical o anotación morfosintáctica, consiste en indicar para cada palabra su categoría gramatical junto con sus rasgos gramaticales más relevantes (Moreno, *et al.*, 2001).

El analizador morfológico de Freeling utiliza un conjunto de etiquetas para representar la información morfológica de las palabras. Este conjunto de etiquetas se basa en las etiquetas propuestas por el grupo EAGLES (del inglés Expert Advisory Group on Language Engineering Standards) para la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas (Padro, 2015). Un ejemplo de esta anotación gramatical, puede verse en la *Figura 2*.

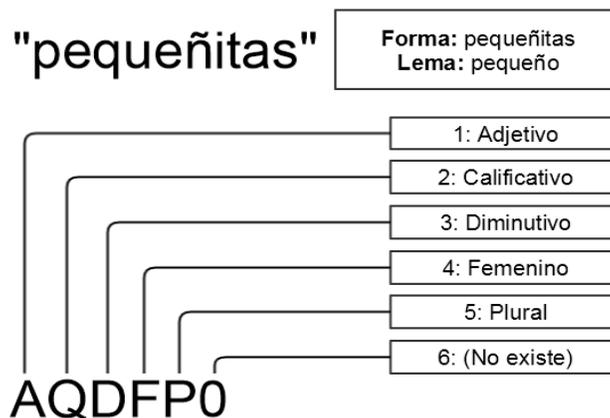


Figura 2. Anotación gramatical.

#### Cuadro 1. Ejemplo de una sección del texto de entrada

---

:si a01 es a02 y a02 es a03 entonces a01 es a03  
 :si a01 es adyacente a a02 entonces a02 es adyacente a a01  
 jugador1 tiene las piezas rojas  
 jugador2 tiene las piezas blancas  
 a1 es una casilla  
 a1 es adyacente a a2 y b1  
 ficha01 es una pieza negra  
 ficha02 es una pieza negra

---

Después de estos pasos, se tiene como salida una lista de oraciones. A su vez, para cada oración, se tiene una lista de posibles etiquetas morfosintácticas, cada una tiene un grado de ambigüedad. Quiroga-Clare

(Quiroga-Clare, 2003) define la ambigüedad como algo que puede ser entendido de dos o más posibles maneras, y hace hincapié en su importancia en PLN. Al problema de eliminar la ambigüedad, se le conoce como desambiguación. El grado de ambigüedad de cada una de las oraciones en la lista, es asignado por el módulo de desambiguación de Freeling y puede variar desde valores cercanos a 0 para etiquetas improbables, hasta valores cercanos a 1 para etiquetas de las que se tiene mucha seguridad que coinciden. Estos valores de probabilidad son asignados por el módulo de Freeling de acuerdo a estudios estadísticos de ciertos corpus. En el paso siguiente, es necesario eliminar las posibles ambigüedades. Se seleccionan únicamente las etiquetas que tienen la mayor probabilidad de ser las correctas.

**Cuadro 2.- Contenido de la base de datos del diccionario para *las* y *hay***

---

```

<entrada>
<palabra>las</palabra>
<significado/>
<morfo>da0fp0</morfo>
<lema>el</lema>
<nsm/>
<bnsm>>false</bnsm>
<identificador>0</identificador>
</entrada>
<entrada>
<palabra>hay</palabra>
<significado>que algo existe</significado>
<morfo>vmip3s0</morfo>
<lema>haber</lema>
<nsm>haber</nsm>
<bnsm>>false</bnsm>
<identificador>0</identificador>
</entrada>

```

---

En caso de que dos etiquetas tengan la misma probabilidad, se escoge la sugerida por Freeling, esto da como resultado una oración con palabras y sus respectivas categorías gramaticales como se ve en el *Cuadro 2*. Es posible que una palabra sea idéntica a otra en sus etiquetas gramaticales, pero que aun así tengan significados distintos. En este caso se utiliza un diccionario que contiene los significados posibles para esa etiqueta. Cabe mencionar que, en los pasos anteriores, varios significados posibles son descartados. Esto puede provocar que un significado incorrecto sea seleccionado si la redacción del texto es tal que el desambiguador de Freeling le asigne menos probabilidad al significado que se quería especificar.

Una palabra identificada con su etiqueta gramatical puede tener más de un significado, lo cual altera el significado de la frase. Para evitar esto, el paso siguiente es mostrar al usuario los significados de las palabras, en caso de que tenga más de uno. Si solo es uno, el programa salta toda esta verificación y se va al paso de conversión a NSM. En los casos en que hay más de un significado se muestran las opciones disponibles al usuario, con el default resaltado. En caso de que la palabra no tenga un significado registrado, o que haya un significado que no está considerado para esa palabra, se le puede indicar al programa que se agregará uno para poder seleccionarlo después.

### Conversión a NSM

Una vez que el texto de entrada está etiquetado, debe ser convertido a un vocabulario formado por términos del Metalenguaje Semántico Natural (NSM). Para esto, se considera que la conversión NSM de una determinada oración, es válida también para otra oración con una estructura sintáctica idéntica. Con esto, se construye una base de datos que relaciona una estructura sintáctica con las instrucciones para convertirla a NSM. Esta estructura sintáctica no es más que la lista de etiquetas gramaticales que corresponden a cada palabra de la oración (*Cuadro 3*). Cuando se encuentra una estructura sintáctica que no coincide con ninguna en la base de datos, se pregunta al usuario si desea agregarla, y se procede de acuerdo.

Las instrucciones de conversión a NSM deben construirse de manera manual, lo cual requiere una inversión de tiempo para cualquier estructura nueva. Las oraciones de entrada cuya estructura sintáctica coincida con las ya existentes en la base de datos, son procesadas inmediatamente, por lo que la información que se pide al usuario es altamente reutilizada. En las instrucciones de conversión, arbitrariamente se toman las letras X, Y, y Z para representar los sujetos o tema principal en las oraciones, mientras que las letras A, B, C, etc., se usan para cualquier otra variable.

**Cuadro 3. Ejemplos de entradas en la base de datos de plantillas**

---

```

<plantilla>
  <patrón>Z VSIP3S0 AQ0CS0 SPS00 Z CC Z</patrón>
  <instrucciones>A ser adyacente B
  A ser adyacente C</instrucciones>
  <variables>A=1
  B=5
  C=7</variables>
</plantilla>
<plantilla>
  <patrón>CS Z VSIP3S0 AQ0CS0 SPS00 Z RG Z VSIP3S0 AQ0CS0 SPS00 Z</patrón>
  <instrucciones>si A ser adyacente B entonces B ser adyacente A</instrucciones>
  <variables>A=2
  B=6</variables>
</plantilla>

```

---

Las instrucciones consisten únicamente de sustituciones, pero esto de manera repetitiva permite procesar oraciones de mayor complejidad.

### Administración de las bases de datos

Existen tres bases de datos: el diccionario de palabras, con el que diferentes palabras que cuentan con la misma etiqueta morfosintáctica pueden ser diferenciadas, el diccionario de términos NSM, en el que cada significado cuenta con una definición en términos del vocabulario de NSM, y la base de datos de estructuras, que se usa para la conversión a NSM. Estas bases de datos pueden construirse desde cero en el programa, o bien pueden cargarse desde archivos XML. El programa se encarga de validar el contenido de las bases de datos antes de cargarlas.

El usuario alimenta información de las palabras nuevas y el resultado se traduce a NSM mediante una tabla de conversiones. Una conversión NSM de determinada oración se considera válida para otra oración si ambas cuentan con una estructura sintáctica idéntica. Las instrucciones para convertir a NMS deben

hacerse manualmente para cualquier estructura nueva. La información leída de las reglas, el estado inicial del tablero, y la interacción con el usuario se guarda en una red semántica, en la cual una oración es válida si su representación NSM existe en la red. También tiene un algoritmo para rechazar cambios que introducen contradicciones o ciclos en el sistema.

## RESULTADOS Y DISCUSIÓN

En la implementación, el usuario da las reglas del juego de *gato*, describiendo el estado inicial y las condiciones para decidir un ganador o un empate. El programa lee las reglas y pregunta al usuario por más información en caso de que no reconozca una palabra o una estructura sintáctica y después realiza análisis sintácticos y semánticos a estos datos. Se usaron fichas blancas y negras como marcadores para distinguir a los dos jugadores, y para el tablero se usó una cuadrícula de 3x3, marcadas con letras para las columnas y números para las filas, como se puede ver en la *Figura 3*. Después una etapa de desambiguación basada en grafos se seleccionan las ramas más probables del análisis de salida. La herramienta lingüística Freeling se usa para estos pasos de análisis y desambiguación.

<b>a3</b>	<b>b3</b>	<b>c3</b>
<b>a2</b>	<b>b2</b>	<b>c2</b>
<b>a1</b>	<b>b1</b>	<b>c1</b>

**Figura 3.** Cuadrícula de 3x3.

### Cuadro 4. Ejemplo de la salida del programa

---

```

>Cargando base de datos NSM... OK.
>Creando motor de inferencia y cargando términos NSM... OK.
>Integrando diccionario semántico y NSM al motor... OK.
> - Integración a la red semántica -
> -- "si a01 ser a02 y a02 ser a03 entonces a01 ser a03", aplicando cambios a la red.
>La regla fue integrada y procesada
> -- "si a01 ser adyacente a02 entonces a02 ser adyacente a01", aplicando cambios a la red.
>La regla fue integrada y procesada
> * "a1 ser adyacente a2" genera: ("a2 ser adyacente a1")
> * "a1 ser adyacente b1" genera: ("b1 ser adyacente a1")
> * "a2 ser adyacente b2" genera: ("b2 ser adyacente a2")
> [ '?' = pregunta, sin signo = orden, vacío = salir ]
Por favor, a continuación haga su solicitud:
> 1: "pieza blanca esta sobre a1".
>La plantilla existe.
>VERDADERO
> [ '?' = pregunta, sin signo = orden, vacío = salir ]
Por favor, a continuación haga su solicitud:
> 2: "pieza negra esta sobre a3".
>La plantilla existe.

```

---

---

>VERDADERO  
> [ '?' = pregunta, sin signo = orden, vacío = salir ]  
Por favor, a continuación haga su solicitud:  
> 3: "pieza blanca esta sobre b2".  
>La plantilla existe.  
>VERDADERO  
> [ '?' = pregunta, sin signo = orden, vacío = salir ]  
Por favor, a continuación haga su solicitud:  
> 4: "pieza negra esta sobre b2".  
>La plantilla existe.  
>FALSO  
>contradicción  
>si a01 no ser a02 y a02 esta sobre a03 entonces a01 no esta sobre a03  
> [ '?' = pregunta, sin signo = orden, vacío = salir ]  
Por favor, a continuación haga su solicitud:  
> 5: "pieza negra esta sobre b3".  
>La plantilla existe.  
>VERDADERO  
> [ '?' = pregunta, sin signo = orden, vacío = salir ]  
Por favor, a continuación haga su solicitud:  
> 6: "pieza blanca esta sobre c3".  
>La plantilla existe.  
>  
>FIN:  
>si pieza blanca esta sobre a1 y pieza blanca esta sobre b2 y pieza blanca esta sobre c3 entonces jugador1 gana.

---

Finalmente, la información y las reglas de la red semántica se usan para deducir e integrar cualquier información nueva. Un pequeño ejemplo de una consulta de información se puede ver en el Cuadro 4. En este punto, aún se ve mucho del procesamiento interno en la salida. Al principio de la interacción el archivo de texto conteniendo las reglas ya ha sido integrado a la red semántica del sistema experto y la entrada del usuario es procesada.

## CONCLUSIONES

Se implementó un convertidor de español a Metalenguaje Semántico Natural y se usaron las herramientas de una librería lingüística para generar oraciones no ambiguas. La conversión conlleva mucho trabajo manual que podría ser automatizado. La salida está formada de términos relativamente sencillos que se usaron para alimentar un sistema experto. Considerar las oraciones como desambiguadas fue muy útil en la depuración y en la predicción de la salida del programa. Se probó el prototipo con reglas del juego de *gato*, se validaron estados del tablero y se modificó el tablero de acuerdo a los movimientos indicados por el usuario.

Se planea extender las capacidades del sistema para ser capaz de entender instrucciones más complejas y tomar ventaja de optimizaciones en el motor de inferencia, así como también añadir cambios en el sistema tales que se puedan usar oraciones imperativas que invaliden información pasada (movimientos anteriores).

Con estos cambios, se podrían asimilar instrucciones más complejas como las del ajedrez, damas inglesas o incluso de ensamblaje de algún objeto.

### LITERATURA CITADA

- Adeli, H., 2003. *Expert systems in construction and structural engineering*. s.l.:CRC Press.
- Agirre, E. & Edmonds, P., 2007. *Word sense disambiguation: Algorithms and applications*. s.l.:Springer Science & Business Media.
- Brezillon, P., 1999. Context in Artificial Intelligence: A survey. *The Knowledge Engineering Review*, 14(01), pp. 47-80.
- Brezillon, P. a. P. J.-C., 1996. Misuse and nonuse of knowledge-based systems: The past experiences revisited. In: *Implementing Systems for Supporting Management Decisions*. s.l.:Springer, pp. 44-60.
- Castillo, E., Gutierrez, J. M. & Hadi, A. S., 1997. Sistemas expertos y modelos de redes probabilísticas. *Academia de Ingenieria*.
- Coppin, B., 2004. Artificial Intelligence Illuminated. In: *Expert Systems*. s.l.:s.n., p. 768.
- Cutting, D. a. K. J. a. P. J. a. S. P., 1992. *A Practical Part-of-Speech Tagger*. s.l., s.n.
- Davis, R. a. K. J. J., 1984. The origin of rule-based systems in AI. *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*.
- Di Eugenio, B., 1992. Goals and Actions in Natural Language Instructions.
- Di Eugenio, B. & White, M., 1992. *On the Interpretation of Natural Language Instructions*. Nantes, France, s.n., p. 55.
- Dominey, P. F., Mallet, A. & Yoshida, E., 2007. Progress in programming the HRP-2 humanoid using spoken language. *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2169-2174.
- Goddard, C. et al., 2006. Semantic primes and universal grammar, Empirical evidence from the Romance languages. In: *Meaning and Universal Grammar: Theory and empirical findings*. s.l.:John Benjamins Publishing Company.
- Harrington, B. a. C. S., 2007. Asknet: Automated semantic knowledge network. *Proceedings of the National Conference on Artificial Intelligence*, Volume 22, p. 889.
- Jackendoff, R., 1990. Semantic Structures. Current studies in linguistics series. *MIT Press Cambridge, MA*.
- Jayas, D. S., White, N. D. & Muir, W. E., 1994. *Stored-grain ecosystems*. s.l.:CRC Press.
- Kornai, A., 2010. *Lecture Notes in Computer Science*. s.l., s.n.
- Linckels, S. & Meinel, C., 2011. Natural Language Processing. In: *E-Librarian Service*. s.l.:Springer, pp. 61-79.
- Malrieu, J.-P., 2002. Evaluative semantics: Cognition, language and ideology. *Routledge*, Volume 3.
- Moreno, J. L. A. et al., 2001. Aspectos ortograficos, lexicos y morfosintacticos del etiquetado linguistico de un corpus de informatica en lengua gallega. *Procesamiento del Lenguaje Natural*, Volume 27, pp. 13-20.
- Padro, L., 2012. Analizadores Multilingues en Freeling. *Linguamatica*, Volume 3, pp. 13-20.
- Padro, L., 2015. *Introduccion a las etiquetas EAGLES*. [Online] Available at: <http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html> [Accessed 31 October 2015].
- Piantadosi, S. T., Tily, H. & Gibson, E., 2012. The communicative function of ambiguity in language. *Cognition*, Volume 122, pp. 280-291.
- Quiroga-Claire, C., 2003. Language Ambiguity: A Curse and a Blessing. *The Translation Journal*, Volume 7.
- Schank, R. C., 1972. Conceptual dependency: A theory of natural language understanding. In: *Cognitive Psychology*. s.l.:s.n., pp. 552--631.

- Sethi, R., 1992. *Lenguajes de programación: conceptos y constructores*. s.l.:s.n.
- Sharples, M. a. H. D. a. H. C. a. T. S. a. Y. D., 1989. *Computers and thought: A practical introduction to artificial intelligence.. The MIT Press*.
- Sowa, J. F., 2015. *Semantic Networks*. [Online] Available at: <http://www.jfsowa.com/pubs/semnet.htm> [Accessed 7 July 2016].
- Sowa, J. F., 2016. *Semantic Networks*. [Online] Available at: <http://www.jfsowa.com/pubs/semnet.htm> [Accessed 08 June 2016].
- Steels, L., 2001. Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5), pp. 16-22.
- Stock, K. et al., 2011. Multilingual Natural Language Spatial Querying using Natural Semantic Metalanguage. p. 391.
- Vásquez, A. C., Quispe, J. P. & Huayna, A. M., 2009. Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática*, 6(2), pp. 45-54.
- Wyner, A., Angelov, K., Barzdins, G. & Damljanovic, D., 2010. On controlled natural languages: Properties and prospects. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Volume 5972, pp. 281-289.
- Yoon, K.-J., 2004. *Korean maum vs. English heart and mind: Contrastive semantics of cultural concepts*. s.l., s.n.
- Zamblera, F., 2010. *Computational NSM: a PROLOG-based notation*. [Online].

## SÍNTESIS CURRICULAR

### **León Verdín Armenta**

Ingeniero en Mecatrónica egresado del Centro de Enseñanza Técnica Industrial (CETI) en 2005, obtuvo su grado de maestro en sistemas computacionales en el Instituto Tecnológico Superior de Zapopan en 2016. Actualmente labora en IBM como desarrollador de una herramienta de pruebas de estrés de hardware en servidores, entre sus áreas de interés se encuentran la lingüística computacional y la inteligencia artificial.

### **Miriam Díaz Rodríguez**

Ingeniera en computación por el Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) en 2007, obtuvo su grado de maestra en ciencias en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV) Unidad Guadalajara en 2010, donde realiza actualmente su doctorado. Es profesora titular en el Instituto Tecnológico José Mario Molina Pasquel y Henríquez, entre sus áreas de interés se encuentran sistemas de eventos discretos y teoría computacional.