UN SISTEMA TUTORIAL INTELIGENTE PARA EL TRATAMIENTO DE LOS ERRORES GRAMATICALES DEL ESPAÑOL COMO LENGUA EXTRANJERA CON FINES ACADEMICOS[1]

ANITA FERREIRA CABRERA
GABRIELA KOTZ GRABOLE
Universidad de Concepción

RESUMEN

Una de las más notables contribuciones de la Inteligencia Artificial (del inglés, *Artificial Intelligence*, (AI)) a la enseñanza-aprendizaje de lenguas, y educación en general, han sido los Sistemas Tutoriales Inteligentes (STI) (del inglés, *Intelligent Tutorial Systems* (ITS)). Estos sistemas están basados en los desarrollos de la Inteligencia Artificial, principalmente, en lo que compete a la utilización de técnicas de comprensión de lenguaje natural (del inglés, *Natural Language Understanding*) o de generación de lenguaje (del inglés, *Natural Language Generation*). Un STI es un programa para la

[1] La investigación que se presenta en este artículo se circunscribe en el contexto del Proyecto FONDECYT 1080165. Modelo *Blended Learning* basado en el enfoque por tareas y aprendizaje cooperativo para la enseñanza del Español como Lengua Extranjera. CONICYT- CHILE

enseñanza-aprendizaje basado en el computador cuya finalidad última es la facilitación de los procesos de aprendizajes personalizados. La etiqueta inteligente se refiere a la capacidad del sistema para analizar gramaticalmente la entrada y luego generar un *feedback* adecuado para el error del estudiante (Heift, 2003; Toole & Heift, 2002). Para ello, se deben implementar técnicas de procesamiento de lenguaje natural (NLP) que se basan en teorías gramaticales para procesar la entrada del estudiante con el objeto de generar una estrategia de *feedback*.

El aumento por el interés en el aprendizaje del Español como LE con fines profesionales y académicos conlleva a la creación de ambientes de aprendizaje más enriquecidos por profesionales e investigadores que apoyen de manera efectiva los procesos tendientes a la autonomía en el aprendizaje. En materia de enseñanza de lenguas extranjeras, se ha corroborado empíricamente el interés en el aprendizaje que despierta por parte de los alumnos la incorporación de plataformas *e-learning* habilitadas con Tecnologías de Información y Comunicación TIC. Sin embargo, al mismo tiempo, se ha observado las limitaciones de dichas plataformas en el tratamiento gramatical específicamente en el reconocimiento de los diferentes tipos de errores de lengua (gramaticales, léxicos, etc.) que cometen los alumnos y la precariedad y poca efectividad en el tipo de retroalimentación en el tratamiento de dichos errores. Los errores y el *feedback* correctivo son parte natural del aprendizaje de la lengua. En consecuencia, en este artículo se plantea como objetivo principal atender a la problemática de mejorar los procesos de aprendizaje del Español como LE con fines profesionales y académicos, a través del *Diseño e implementación de un Sistema Tutorial Inteligente para el Español como LE.* En el diseño se considera como aspectos principales el aprendizaje de la lengua en contextos académicos universitarios, además se sustenta en los principios metodológicos del enfoque por tareas y cooperativo que garantizan ambientes psicolingüísticamente óptimos para el aprendizaje de lenguas por medio de la tecnología.

1. Introducción

Uno de los grandes aportes que realizó la Inteligencia Artificial (del inglés, *Artificial Intelligence, AI)* a la enseñanza-aprendizaje de lenguas han sido los Sistemas Tutoriales Inteligentes (STI) para Lenguas Extranjeras (LE) (del inglés, *Intelligent Tutorial Systems (ITS) for Foreign Language (FL).*

Estos sistemas reciben el nombre de inteligentes porque son capaces de comportarse como un experto, ya que pueden analizar la situación en la que se encuentra el estudiante y proporcionar una solución acorde a la problemática. Para ello, se deben implementar técnicas de procesamiento de lenguaje natural (del inglés, *Natural Language Processing, NLP*) que se basan en teorías gramaticales para procesar la entrada del usuario con el objeto de generar una respuesta.

Este estudio se plantea como objetivo principal el diseño y la implementación de un analizador automático de errores gramaticales para el español en el contexto de un tutorial inteligente para la enseñanza del Español como Lengua Extranjera: *Parser* ELE-UDEC.

La particularidad de este analizador gramatical de entradas (*parser*) es que debe procesar entradas erróneas y para ello, es necesario poder predecir los errores que puede cometer un usuario en un momento determinado de su aprendizaje y en un tópico gramatical específico.

El analizador sintáctico está concebido para apoyar la focalización en las formas gramaticales que requieren precisión en el manejo de la lengua con fines académicos.

El objetivo principal es mejorar la competencia lingüística que los estudiantes necesitan para insertarse en el ámbito académico, ya sea a nivel de pre o post-grado en la Universidad de Concepción.

2. Objetivos

Este estudio se plantea como objetivo principal el diseño y la implementación de un analizador automático de errores gramaticales para el español en el contexto de un tutorial inteligente para la enseñanza del Español como Lengua Extranjera (ELE).

Con el objeto de diseñar un Sistema Tutorial Inteligente, es decir un sistema que incorpore técnicas de PLN que sea capaz de reconocer de manera efectiva los errores gramaticales que efectúen los alumnos y dar una respuesta útil y específica, es necesario construir un analizador sintáctico *(parser)* para el tratamiento de dichos errores que opere en un entorno virtual y que dé respuestas automáticas. Otra característica importante de este *parser* es que dará respuestas diferentes según el nivel de aprendizaje en que se encuentre el alumno.

3. Marco teórico

3.1. ¿Qué es un analizador automático?

Un analizador automático (*parser)* de lenguaje natural toma un texto en lengua escrita como entrada y produce una representación formal de la estructura sintáctica (y, a veces semántica) del mismo, generalmente en forma de árboles.

Los *parsers* diseñados para la enseñanza de lenguas generalmente contienen un componente que anticipa o detecta los errores en el caso de que las reglas gramaticales sean violadas.

Por ejemplo, un procedimiento muy común para la detección de errores es la incorporación de una gramática de errores con *buggy rules* (reglas agramaticales), lo que permite que el *parser* procese una oración que contiene uno o más errores e identificarlos.

3.2. Utilidad de los *parsers*

La utilidad de los *parsers* sintácticos en la enseñanza de lenguas fue analizada empíricamente. Jouzulynas (1994, en Heift y Schulze 2007) evaluó la utilidad de los *parsers* sintácticos en el diagnóstico de errores. Para ello, analizó errores en un corpus de aprox. 400 páginas de ensayos en alemán escritos por estudiantes norteamericanos en cursos de segundo año de lengua. Su estudio demuestra que el área de aprendizaje más problemático es la sintaxis, seguido por la morfología. El estudio, que fue realizado con la ayuda de un *parser* del alemán, indicó que la mayoría de los errores de los alumnos (80%) no son de origen semántico y, por lo tanto, pueden ser reconocidos por un *parser* sintáctico.

Holland, Sams y Kaplan (1995) responden a la pregunta *¿Para qué sirven los parser?* basados en los excelentes resultados obtenidos con BRIDGE, un sistema de *CALL* (del inglés, *Computer Assisted Language Learning)* basado en *parser* utilizado con personal militar norteamericano que aprendía alemán.

Estas conclusiones son confirmadas con pruebas realizadas en otras aplicaciones de *CALL* basadas en *parser.*

Por ejemplo, Nagata (1996) comparó sistemas de *CALL* con los sistemas de *ICALL (Intelligent-CALL*) o *CALL basado en parser,* y concluyó que *ICALL* resulta más eficiente, ya que realiza un uso más efectivo del computador, en el sentido de proveer una respuesta inmediata y apropiada para el alumno. Para ello utilizó el sistema Nihongo-CALI para el aprendizaje del japonés.

Estas conclusiones representan una sólida evidencia de los beneficios de la inclusión de técnicas de procesamiento de lenguaje en *CALL.*

3.3. NLP

El procesamiento de lenguaje natural (del inglés, *Natural Language Processing, NLP*), es una rama de la Inteligencia Artificial (*AI*), pero la *AI* no trata sólo con el procesamiento de lenguaje natural. En general, la *AI* se

ocupa de emular aspectos de la inteligencia humana a través de máquinas. La *AI* se ocupa de representaciones del conocimiento y la investigación y aplicación de tales algoritmos y técnicas en juegos, máquinas de aprendizaje, robótica.

NLP no sólo es una importante rama de la *AI,* sino también de la Lingüística computacional.

La Lingüística computacional es una rama de la Lingüística que apunta al desarrollo de modelos computacionales para la lengua tanto escrita como hablada. Tradicionalmente se enfoca al desarrollo de *HLT (Human Language Technology)*, pero en la actualidad se ha dirigido más hacia al análisis detallado de problemas prácticos.

La tecnología NLP permite programar el computador con suficiente información lingüística en forma de reglas y patrones que permite realizar numerosas actividades relacionadas con el aprendizaje de lenguas. Por ejemplo, puede analizar estructuras de oraciones entregadas por los usuarios, detectar errores lingüísticos y, en algunos casos, aparentar comprensión entregando respuestas *ad hoc* o respondiendo a una conversación.

Por ejemplo, puede analizar estructuras de oraciones entregadas por los usuarios, detectar errores lingüísticos y, en algunos casos, aparentar comprensión entregando respuestas *ad hoc* o respondiendo a una conversación.

Los sistemas con tecnología NLP les dan a los alumnos la factibilidad de crear oraciones nuevas y originales en la lengua que están aprendiendo, ingresarlas al computador y recibir una respuesta como retroalimentación. Esto es lo que diferencia *ICALL* de *CALL*.

NLP se divide en dos partes: en la comprensión de lenguaje natural (*Natural Language Understanding, NLU*) y en la generación de lenguaje natural (*Natural Language Generation, NLG).*

3.4. Técnicas básicas para el tratamiento de lenguaje natural

La primera etapa de cualquier sistema de procesamiento de la lengua tiene lugar en el nivel textual. En este nivel, el texto puede ser considerado como una simple secuencia de caracteres. Las tareas básicas que deben abordarse a este nivel son: a) la segmentación del texto, b) el filtrado de información no relevante y c) la localización de unidades tratables. A pesar de su aparente simplicidad, estas tareas pueden presentar algunas complicaciones.

a) Segmentación del texto. El texto debe ser segmentado en fragmentos tratables de manera automática. La dificultad de la tarea depende tanto de las características de los fragmentos a obtener (párrafos, oraciones, etc.), como de la fuente de la cual se obtienen (texto marcado, texto plano, resultado de una transcripción a partir de voz, etc.). Si se desea segmentar un texto en párrafos y se dispone de signos de puntuación y se puede distinguir mayúsculas y minúsculas, entonces la tarea es relativamente sencilla (aun cuando un signo de puntuación puede cumplir funciones diferentes de la de separación; por ejemplo, un punto puede formar parte de un nombre propio, de una sigla, de una fórmula o de un acrónimo). Si deseamos segmentar un texto en oraciones, la tarea es obviamente más difícil y será necesario algún tipo de conocimiento lingüístico, lo mismo que si la fuente a tratar es producto de una transcripción y carece de signos de puntuación y de distinción entre mayúsculas y minúsculas.

Esta tarea es realizada por un analizador léxico que crea estos segmentos, también llamados *tokens* y son estos *tokens* los que son procesados por el analizador morfológico, para posteriormente realizar el análisis sintáctico para construir luego la estructura de datos, por ejemplo un árbol de análisis o árboles abstractos de sintaxis.

b) Filtrado de información no relevante. Los textos a tratar vienen a menudo acompañados de otros elementos que deben ser eliminados o extraídos para facilitar el tratamiento. Así, si la fuente de información es una página de Internet, junto a los fragmentos de texto tratables, aparecen

diferentes tipos de marcas que definen las características de visualización de la página, enlaces con otras páginas o dentro de la misma página, objetos no textuales, como por ejemplo imágenes, animaciones, tablas, gráficos, etc. Si el texto está marcado de forma consistente, el filtrado es relativamente sencillo, pero, a menudo no es así y la dificultad de la tarea aumenta. Otra tarea es la de la identificación de la lengua en los casos en que se desconozca *a priori* la lengua del texto, por ejemplo si el texto ha sido recuperado a través de Internet, o en los casos en que un texto contenga fragmentos correspondientes a lenguas diferentes.

c) Localización de unidades tratables. Las unidades básicas de tratamiento son las palabras. Localizar las palabras ortográficas es sencillo si el espacio o los signos de puntuación actúan como separadores. En las lenguas en que esto no es así, por ejemplo en el japonés, o en los casos en que la puntuación no existe, el problema es mayor. Pero aún en los casos en que se hayan localizado las palabras ortográficas existen casos problemáticos. Por ejemplo:

- Distinción entre palabras ortográficas y palabras gramaticales. Por ejemplo, en la conjunción *sin embargo* en que una palabra gramatical corresponde a dos ortográficas o *dímelo* en que una palabra ortográfica corresponde a tres palabras gramaticales, o contracciones como *del* o *al*.
- Términos multipalabra, como en el caso de *San Pedro* o *Buenos Aires*
- Fechas, fórmulas, siglas, abreviaturas, etc.
- Nombres propios (de persona, geográficos, etc.).
- Palabras desconocidas, neologismos o errores. Es decir, palabras que no figuran en los diccionarios disponibles.

El uso de diccionarios específicos o terminológicos que complementen los diccionarios generales, de procesadores para tratar las unidades no estándar (por ejemplo extractores de fechas o identificadores de

nombres propios), y de técnicas estadísticas y de aprendizaje automático contribuye a solucionar estos problemas.

3.4.1 Análisis morfológico

El siguiente paso en el tratamiento de la lengua consiste en el análisis morfológico. Esta tarea es normalmente realizada por un analizador morfológico cuyo papel es el de recuperar la morfología de las palabras, es decir, las formas con que se construyen las palabras a partir de unidades significativas más pequeñas, llamadas *morfemas.* Los morfemas se clasifican en dos clases: troncos (*stem*) y afijos. Generalmente, las palabras se forman a través de mecanismos de flexión, derivación o composición a partir de sus formas básicas o troncos. Esta tarea se denomina *stemming* o lematización.

El análisis morfológico de las formas flexivas del español es relativamente sencillo, ya que la flexión responde a patrones bastante regulares. El número de sufijos flexivos es de unos 200 y reglas de combinación ascienden a unas 500. En cambio, la derivación o la composición son más complicadas y suelen venir combinadas con la flexión.

A menudo, sobre todo para lenguas con poca complejidad morfológica, como la inglesa, el analizador morfológico se reduce a un *formario*, es decir a un diccionario de formas completas. Si el *formario* está correctamente implementado, la eficiencia del proceso de análisis es alta. Por otra parte, los *formarios* son fácilmente extensibles, soportan entradas multipalabra y es posible su construcción a partir de generadores morfológicos.

En otras ocasiones, la representación explícita de todas las formas no es conveniente y se debe recurrir a un proceso de análisis morfológico en el momento del procesamiento de la oración. Para el caso del español, dado su alta capacidad flexiva (sobre todo en los verbos) un analizador morfológico que actúe en el momento del tratamiento debe sustituir o completar el *formario.*

3.4.2. Etiquetado de partes del habla (*POS tagging*)

A cada una de las unidades léxicas presentes se le debe asignar el conjunto de sus categorías gramaticales posibles. A este proceso se lo denomina *etiquetado* o *tagging.* El problema es que las palabras tomadas en forma aislada son ambiguas respecto de su categoría. Si se considera el siguiente ejemplo: «Yo bajo con el hombre bajo a tocar el bajo bajo la escalera». La palabra *bajo* puede pertenecer, dependiendo del conjunto de etiquetas que se manejen, a un mínimo de cuatro categorías diferentes: verbo, adjetivo, nombre y preposición. El analizador morfológico devolverá toda ellas para cada una de las apariciones de la forma *bajo* en la oración. Afortunadamente la categoría de la mayoría de las palabras no es ambigua dentro de un contexto. Para un humano es relativamente simple eliminar la ambigüedad en la categorización, lo hace de manera rápida y eficiente, pero no lo es para una máquina. Para ello existen los desambiguadores morfosintácticos (*POS taggers*) o etiquetadores, cuya misión es la de realizar automáticamente esta tarea.

El objetivo de un etiquetador es el de asignar a cada palabra la categoría más *apropiada* dentro de un contexto. Por supuesto, la calidad de este dependerá del grado de precisión (*granularidad*) del etiquetado, del contexto considerado y de la información de que se disponga para considerar *apropiada* una etiqueta o secuencia de etiquetas. A veces, los etiquetadores no resuelven totalmente el problema de la ambigüedad gramatical y se limitan a eliminar las opciones imposibles o menos probables. Este es el caso de los denominados *desambiguadores reduccionistas*.

Existen tres grandes tipos de etiquetadores o métodos de etiquetado: los basados en reglas, los estadísticos o probabilísticas y los híbridos basados en transformaciones. Los *etiquetadores basados en reglas* utilizan conocimiento lingüístico *knowledge-driven taggers*), generalmente expresado en forma de reglas o restricciones para establecer las combinaciones de etiquetas aceptables o prohibidas. Las reglas se escriben manualmente, responden a criterios lingüísticos y se representan en forma explícita. Se trata

de sistemas de muy alta precisión (por ejemplo, el EngCG de Karlsson, que implementa las denominadas *Constraint Grammars* para el inglés, comprende un millar de reglas y supera el 99,5 % de corrección). El costo de desarrollo es alto y también lo es el costo de adaptación a otros dominios.

Los *etiquetadores estadísticos* se basan en la evidencia empírica obtenida de corpus lingüísticos voluminosos (*data-driven taggers*). El costo es por ello mucho menor aunque también es menor su grado de precisión, superior en cualquier caso al 97 %, suficiente en algunas aplicaciones. Los sistemas son independientes de la lengua y fácilmente adaptables a otras lenguas y dominios. El problema de estos sistemas reside en el aprendizaje del modelo estadístico utilizado. En este sentido es notable, y creciente, el uso de técnicas de aprendizaje automático. Se han utilizado técnicas de aprendizaje supervisado partiendo de corpus etiquetados manualmente y técnicas de aprendizaje no supervisado en las que no es precisa (o está limitada) esa intervención manual. Se han utilizado modelos muy sencillos del lenguaje, como por ejemplo los bi-gramas en los que la probabilidad de una etiqueta se estima simplemente con el contexto de la etiqueta anterior, o los tri-gramas en los que el contexto abarca a las dos etiquetas precedentes.

En tercer lugar, los *sistemas híbridos* combinan métodos estadísticos y lingüísticos para intentar recoger los aspectos positivos de cada una de ellas y evitar sus limitaciones.

Recientemente han comenzado a utilizarse sistemas de desambiguación por combinación. Se trata de combinación de diferentes modelos del lenguaje en un único desambiguador, de combinación de desambiguadores mediante votación u otros procedimientos más sofisticados (*bagging, boosting*), de aprendizaje iterativo (*bootstrapping*) y otros muchos. (Rodríguez, 2000)

3.4.3. Análisis sintáctico o *parsing*

Una vez analizado morfológicamente y etiquetado o desambiguado total o parcialmente el texto, puede realizarse el análisis sintáctico.

El *parsing* es un proceso por medio del cual se convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada. Un árbol correcto es aquél que cubre todos y solo los elementos del enunciado y en cuyo tope tiene un símbolo S (del inglés *sentence* u oración).

Existen principalmente dos formar de realizar el análisis:

- *Top-Down-Parser*: un analizador puede empezar con el símbolo inicial *S* e ir dividiendo la entrada progresivamente en partes cada vez más pequeñas, hasta llegar a las palabras.
- *Bottom-Up-Parser*: un analizador puede empezar con la entrada e intentar llegar hasta el símbolo inicial.

Existen muchas técnicas de *parsing*. Sin embargo, muchas veces persisten los dos grandes problemas propios de estos analizadores, la ambigüedad y el costo informático que implica el tiempo de procesamiento. Se puede solucionar los problemas de ambigüedad y pensar en un análisis sintáctico en profundidad. Sin embargo, hay ocasiones en que este análisis no es posible o no es conveniente porque es muy lento y/o costoso y se nos presenta la necesidad de realizar un análisis superficial o fragmental en lugar de o como paso previo al análisis sintáctico en profundidad. Además, para muchas aplicaciones no es necesario desarrollar todo el árbol.

Se ha señalado la insuficiencia de los métodos convencionales de análisis sintáctico para tratar textos no restringidos. Problemas como la dificultad de una segmentación adecuada, la obtención de no uno sino varios (a menudo muchos) árboles de análisis o la necesidad de ampliar la cobertura del analizador al tratamiento de oraciones no gramaticales o que incluyan palabras desconocidas tienen difícil solución en el marco tradicional de un analizador sintáctico que trate de obtener un árbol de análisis completo del texto basándose en una gramática de amplia cobertura. Ante ello caben dos aproximaciones: analizar en profundidad pero no todo, caso del *análisis*

fragmental, y analizar todo pero no en profundidad, caso del *análisis superficial*.

El objetivo de los analizadores fragmentales, también denominados agrupadores sintácticos o *chunkers*, es la detección de frases nominales, verbales, adjetivas, adverbiales básicas (sin recursión). A veces se trata simplemente de detectar el segmento (es lo que se denomina *parentizado* o *bracketting*), mientras que en otras ocasiones se desea obtener el etiquetado correcto y la estructura sintáctica del segmento. Es frecuente el uso de técnicas de estados finitos y la actuación de transductores en cascada. También se han utilizado técnicas de aprendizaje automático. A menudo el siguiente paso es la obtención de dependencias y de relaciones sintácticas entre los segmentos. (Rodríguez, 2000).

Otro tipo de *parser* es el *parser superficial* o parcial se denomina *shallow parsing* o análisis ligero. Este se encarga solo de encontrar los componentes principales de la frase como nombres o verbos.

También existe un método llamado *chart parser* que funciona como una tabla con resultados de análisis parciales. Hay dos sistemas muy populares que funcionan con esta metodología: el algoritmo CKY y el de Earley.

La mayoría de los analizadores modernos son al menos en parte *estadísticos*, esto quiere decir que se basan en unos datos de entrenamiento que han sido analizados manualmente. Este enfoque permite al sistema reunir *información* sobre la frecuencia con que ocurren ciertas construcciones en un contexto específico. Algunos de estos enfoques han incluido gramáticas libres de contexto probabilísticas, sistemas de máxima *entropía* y redes *neuronales*.

Los *parser* diseñados para la enseñanza de lenguas generalmente contienen un componente que anticipa o detecta los errores en el caso de que las reglas gramaticales sean violadas. Por ejemplo, un procedimiento muy

común son las *buggy rules* (reglas *malas*), que hacen posible que el *parser* procese una oración que contiene uno o más errores e identificarlos.

Investigadores del área analizaron, con la ayuda de métodos estadísticos, corpus de alumnos de L2 para poder determinar las *buggy rules* necesarias para ser incluidas en el *parser*. La importancia de los *parsers* en CALL fue muy discutida en la última década por varios investigadores: Nagata, Matthews, Holland, Maisano, entre otros (Heift, Trude y Schulze, Mathias, 2007; Schulze, 2008). Holland et al. (1995) se refieren a las posibilidades y limitaciones de tutores de lengua basados en *parser*, realizan una comparación entre el CALL convencional y el *CALL* basado en *parsers* (p.28) y concluyen que en CALL basado en *parser* el estudiante puede escribir una gran variedad de oraciones y desarrollar de una forma relativamente libre habilidades para el mejoramiento de la producción escrita.

El componente de *parse*r consta de un pre-procesador de texto, un analizador morfológico, un diccionario y un *parser* que proporciona el árbol sintáctico (Figura 1)

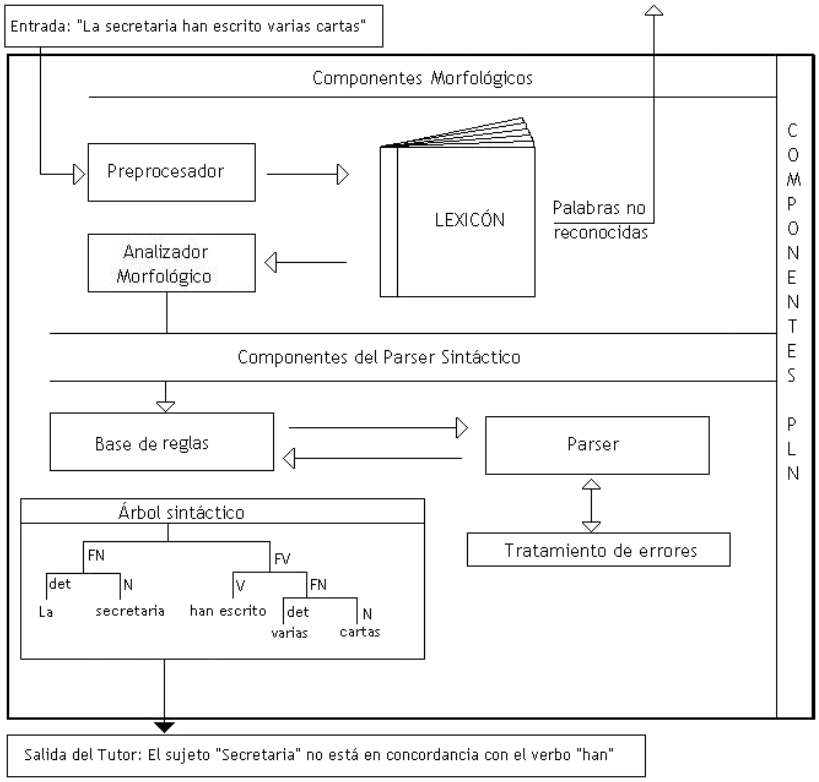

Figura 1: Diagrama simplificado de un componente de *parsing.*

3.5. Los Sistemas Tutoriales Inteligentes y el Aprendizaje de lenguas extranjeras

Uno de los grandes aportes que realizó la Inteligencia Artificial (IA) a la enseñanza-aprendizaje de lenguas han sido los Sistemas Tutoriales Inteligentes (STI) para lenguas extranjeras (LE) (del inglés, *Intelligent Tutorial Systems (ITS) for Foreign Language (FL).* Estos sistemas reciben el nombre de inteligentes porque son capaces de comportarse como un experto, ya que pueden analizar la situación en la que se encuentra el estudiante y proporcionar una solución acorde a la problemática. Para ello, se deben implementar técnicas de procesamiento de lenguaje natural (NLP) que se basan en teorías gramaticales para procesar la entrada del usuario con el

objeto de generar una respuesta (Graesser, A.C., Person, N., Harter, D., y TRG, 2001; Graesser, Jeon, Yang y Cai, 2007).

Los STI constan de tres componentes básicos: 1) *El módulo experto*, donde se almacena el conocimiento del dominio; 2) *el módulo del alumno,* que contiene información acerca del alumno (sus progresos, sus errores, etc.) y 3) *el módulo tutorial,* que contiene técnicas y estrategias de enseñanza.

Los STI para LE han incorporado técnicas de NLP, por ejemplo para el análisis de las entradas en lenguaje natural de los estudiantes o para modelar la competencia de una lengua extranjera. Esto es para que los alumnos /usuarios del sistema tengan una retroalimentación más precisa y flexible. Estos sistemas utilizan técnicas específicas de analizadores sintácticos (*parsing*) para analizar las entradas de los alumnos e identificar los errores que cometen en dichas oraciones. Estas técnicas de NLP han permitido a los sistemas manejar algunas estrategias de *feedback* más sofisticadas como las claves metalingüísticas y los *informes de errores* basados en los análisis de los errores presentados (Ferreira, Moore y Mellish, 2007). Muchos STI parecen haber sido creados sin referencia a los estudios que existen en relación con las habilidades de los estudiantes de lenguas (Holland, Kaplan y Sams 1995, Bull 1997, Chapelle 1997). Creemos que para mejorar la efectividad de dichos sistemas, los investigadores deberían poner mayor atención en los resultados que emergen de la comunidad de Adquisición de Segundas Lenguas. No está claro que podamos formalizar principios tutoriales que sirvan como estructuras de control de un sistema tutorial inteligente sin haber examinado antes las interacciones presenciales en la modalidad tradicional (*face-to-face*) y tutorial (*one-to-one*) (Tomlin, 1995). Es así como muchos estudios empíricos previos en el área de los STI procedimentales han utilizado como modelos las interacciones de un estudiante y su tutor (modalidad *one-to-one*, presencial). Sin embargo, en la enseñanza de lenguas extranjeras, la interacción usualmente toma lugar en ambientes de clases tradicionales (modalidad *face-to-face*, presencial). Por esa razón, hemos llevado a cabo estudios empíricos previos (Ferreira, 2003,

2006, 2007) con el objeto de investigar el tipo, frecuencia y efectividad de las estrategias de *feedback* utilizadas en la enseñanza del Español como Lengua Extranjera en tres contextos de aprendizaje de lengua distintos: un estudio observacional de interacciones profesor-alumno en clases tradicionales presenciales (modalidad *face-to-face*), un estudio de caso de interacciones en clases tutoriales presenciales (modalidad *one-to-one*), y un estudio experimental en el cual los estudiantes interactúan con una aplicación computacional *no presencial* (modalidad *one-to-one* a distancia) accesada en línea a través de la red Internet. Los resultados de estos tres estudios empíricos han resultado ser similares en relación con los tipos, frecuencia, y efectividad de las distintas estrategias de *feedback* correctivo que se investigaron. Se propone, entonces, que los STI para lenguas extranjeras deberían implementar estrategias de *feedback* correctivo que estimulen a los estudiantes a la auto-reparación de sus errores.

Dos ejemplos de STI destacados son el *E-Tutor,* desarrollado por Heift y Schulze (2007) para el alemán (clasifica los errores por medio de una Lista de Prioridad de Error flexible: las limitaciones de gramática pueden ser reordenadas a fin de reflejar el énfasis de una actividad) y el STI *Banzai* (Nagata, 2002) para el japonés (emplea la tecnología NLP para capacitar a los estudiantes para producir libremente frases en japonés y proporcionar un *feedback* detallado en lo que concierne a la naturaleza específica de los errores). La experiencia con el *E-Tutor* apoya la necesidad de un sistema de CALL que considere múltiples errores de acuerdo con la enseñanza de lenguas. El sistema *Banzai* destaca la importancia de determinar generar *feedback* inteligente dirigido a determinadas deficiencias en el rendimiento.

La transformación de CALL a ICALL o CALL basado en *parser* para ser aplicado a la adquisición de lenguas extranjeras exige un cambio de enfoque en la metodología de enseñanza. En efecto, mientras que CALL enfatiza en las tareas y actividades comunicativas, ICALL pone énfasis en las formas lingüísticas (del inglés, *Focus on Form (FonF*)), (Holland, Kaplan y Sams, 1995).

La efectividade este enfoque en las formas lingüísticas, sobre todo en adultos, es ampliamente reportada por Long y Robinson (1998, cit. en Marina Dodigovic, 2005)

Dado que el *parser* de un STI para lenguas extranjeras debe ser capaz de procesar entradas erróneas, las teorías tradicionales de análisis de errores han recobrado vigencia y una nueva perspectiva. «El análisis de errores es el proceso que determina la incidencia, la naturaleza, las causas y las consecuencias de los fracasos lingüísticos» (James 1998:1).

3.5.1. El análisis de los errores

El tratamiento de errores gramaticales ha recibido máxima atención en CALL basado en *parser* o ICALL. La identificación y la descripción de errores juegan un rol importante en la investigación, desarrollo y posterior diseño de *parsers* que pueden procesar entradas erróneas.

La teoría del Análisis Contrastivo (CA) (Lado, 1957) trataba de explicar todos los errores de los alumnos a través de la comparación con su lengua nativa. Este proceso es descrito como transferencia o interferencia. Los seguidores del CA argumentan que la lengua del alumno puede ser *construida* contrastando el sistema de la L1 con el de la L2. Según sus postulados los fenómenos que más difieren entre ambas lenguas son los más difíciles de adquirir y tienen mayor probabilidad de errores (i.e. la flexión de los adjetivos en una lengua y no en la otra). Sin embargo, en aquellos fenómenos que no difieren mucho (i.e. algunas letras tienen el mismo sonido en ambas lenguas), son más fáciles de adquirir y con menos probabilidades de cometer errores.

Sin embargo, se ha comprobado que muchos de los errores que cometen los alumnos no son atribuibles a la interferencia con la lengua materna y, más bien, provienen de errores en la aplicación de las reglas de la lengua meta.

Un importante prerrequisito para la adecuada descripción de los errores y la generación de respuestas útiles en *CALL* es una apropiada

clasificación de los errores. El uso de una clasificación de errores tiene las siguientes ventajas:

- La descripción de errores y sus respuestas pueden estar basadas tanto en errores individuales como en clases de errores
- Los errores que pertenecen a una misma clase pueden ser tratados de manera similar
- La clasificación de los errores permite a los sistemas basados en *parser* no solo guardar los errores sino también calcular el número de errores que pertenecen a una misma clase. Lo que resulta es un *feedback* más efectivo, ya que puede estar basado en un modelo de alumno más claro y comprensible.

4. Diseño e implementación de un analizador para el Español como LE en contextos académicos

La particularidad de este analizador gramatical de entradas (*parser*) es que debe procesar entradas erróneas y para ello, es necesario poder predecir los errores que puede cometer el usuario en un momento determinado de su aprendizaje y en un tópico gramatical específico.

Hoy en día no existe una tipología de errores universal, disponible para cualquier sistema lingüístico que se desee tratar. Esto se debe a que todas las lenguas son diferentes y los tipos de errores identificados pueden variar considerablemente. Así pues, una tipología de errores nunca puede ser elaborada independientemente del objetivo para el cual ha sido concebida.

El objetivo en particular es contrastar la taxonomía de base teórica basada en la literatura especializada con las muestras empíricas resultantes de un estudio observacional en clases tradicionales con el fin de obtener una información más acotada de los errores que podrían llegar a cometer alumnos de ELE con lengua materna inglés en su mayoría.

Este análisis da como resultado una taxonomía de errores acotada a la realidad muestral, pero que debe ser nuevamente ajustada para ser aplicada

al tutorial con características particulares y con un modelo de alumno específico.

Para ello es necesario trabajar además en la selección de ejercicios que sean adecuados para un tutor inteligente. Tenemos que tener en cuenta que el tutor debe *comprender* de manera automática la frase entregada por el usuario, debe realizar el análisis, extraer los errores si los hay y dar una respuesta acorde.

- Elaboración de ejercicios

Se elaboraron una serie de ejercicios teniendo en cuenta el nivel de proficiencia del alumno. Para ello se operó con los descriptores del ALTE para el nivel de experticia con que funcionará el tutor, en este caso, el nivel B2.

También se tuvo en cuenta, no solo las respuestas correctas esperables según el ejercicio en cuestión, sino que también las posibles respuestas incorrectas que tendrá que detectar el analizador. Para ello se tomó como base la taxonomía de errores.

- Etiquetado de partes del habla o *POS Tagging*

Para el correcto funcionamiento del analizador sintáctico debe realizarse previamente o de manera simultánea un análisis morfológico.

Los analizadores morfológicos para el español utilizan una serie de etiquetas para representar la información morfológica de las palabras. Este conjunto de etiquetas se basa en las etiquetas por el grupo EAGLES para la anotación morfosintáctica de lexicones y corpus para las lenguas europeas. Esto es independiente de que haya atributos que no estén especificados y, por lo tanto, haya que crear etiquetas *ad hoc* o, por el contrario, etiquetas que para el corpus no se consideran relevantes y no haya que incluirlas.

- Elaboración de la gramática

Para el funcionamiento del *parser*, es necesario construir una gramática, es decir un conjunto de reglas de formación de frases que el analizador reconocerá. Además, es necesario definir un conjunto de reglas agramaticales (*buggy rules)*, lo que permite que el *parser* procese una oración que contiene uno o más errores e identificarlos.

- Jerarquización de errores

Debido a que el *parser* debe procesar entradas erróneas, un elemento importante es el tratamiento de los errores. Este debe tener especificaciones muy precisas, debe tolerar, detectar y diagnosticar errores, como asimismo ser capaz de entregar una respuesta flexible en el reporte de los mismos.

Un tema importante es el tratamiento de los errores múltiples. Por un lado es deseable que el programa sea capaz de detectar y explicar todos los errores, pero eso no significa que tenga que desplegar cada uno de los errores detectados. La ausencia de un mecanismo de filtro puede llegar a agobiar al alumno. Por ello, nuestro sistema contiene una jerarquía de errores, que ejemplificaremos con el caso de los ejercicios de futuro simple y compuesto:

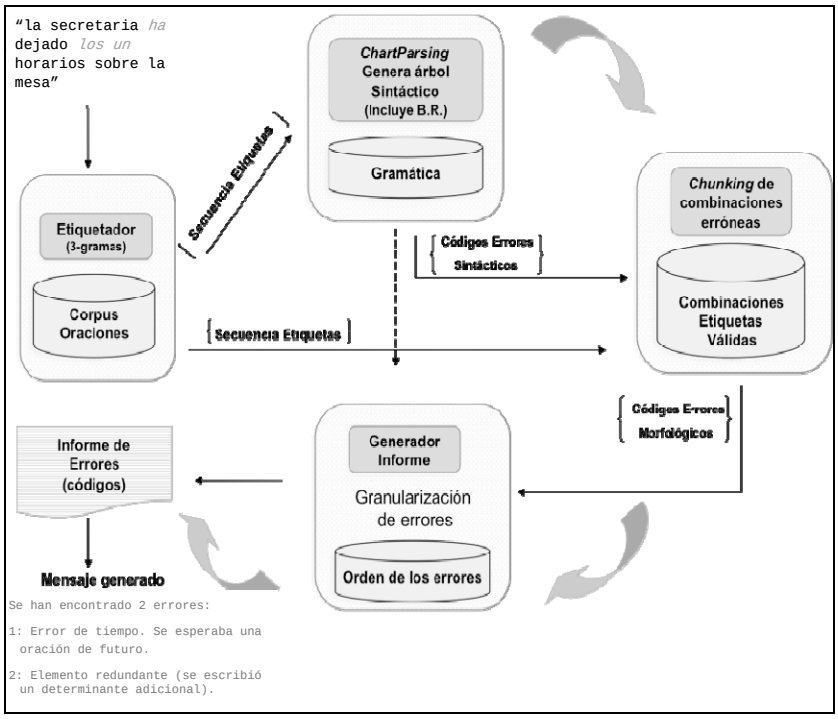

Figura 2: Modelo del *Parser* ELE-UDEC del Sistema Tutorial Inteligente para el Español

En la Figura 2 se precisan las distintas etapas del procesamiento lingüístico de la entrada «*la secretaria ha dejado los un horarios sobre la mesa» realizada por el *Parser* ELE-UDEC. Esto es:

Procesamiento entrada

1: El estudiante ingresa una respuesta al ejercicio presentado por el Sistema Tutorial Inteligente.

2: Se etiqueta la entrada usando un etiquetador que aplica la técnica de n-gramas (en este momento 3-gramas).

3: La oración etiquetada es utilizada para el análisis sintáctico y morfológico.

Análisis Morfológico y Sintáctico

El análisis Morfológico y Sintáctico, se desarrolla de manera independiente.

4. El análisis sintáctico utiliza técnicas de *parsing*, que es una técnica de análisis de gramáticas de contexto libre. En particular, se utiliza *ChartParsing*, que se diferencia de los otros métodos de *parsing* por la utilización de programación dinámica, lo que la hace más eficiente. Esta gramática incluye reglas de errores sintácticos que corresponden principalmente a la estructura de la oración (por ejemplo, errores de orden, de omisión y adición de palabras entre otros, mediante *Buggy Rules*) y de esta forma los errores son detectados mediante la inspección del árbol generado por el *parsing*.

5 .El análisis morfológico utiliza una técnica llamada *chunking*, que sirve para la detección de ciertas identidades o secuencias específicas de texto, de esta forma esta técnica se usa para buscar errores morfológicos dentro de una oración mediante reglas similares a las *Buggy Rules* pero que en este caso se centra en la búsqueda de secuencias de etiquetas específicas dentro de una oración que representa un error morfológico, como por ejemplo: una oración con error de tiempo o número. Los errores morfológicos son informados mediante una codificación del mismo. Actualmente se informa el tipo de error (modo, número, género, etc.), seguido del rasgo esperado, por ejemplo, «error de número, se esperaba una oración escrita en singular».

6. Una vez que se tienen los errores tanto sintácticos como morfológicos, se desarrolla un informe que da cuenta de los errores encontrados de acuerdo a la granularidad de errores definida.

A continuación se muestran algunas pantallas con ejemplos de los análisis sintácticos realizados por el *Parser* ELE-UDEC.

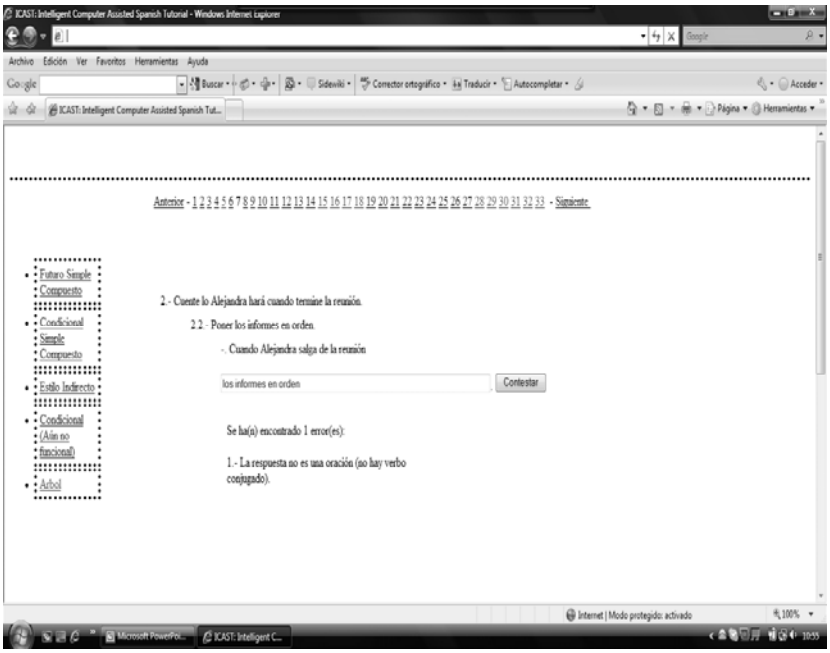

Figura 3: Identificación del tipo de error «Omisión de verbo» realizada por el *Parser* ELE-UDEC

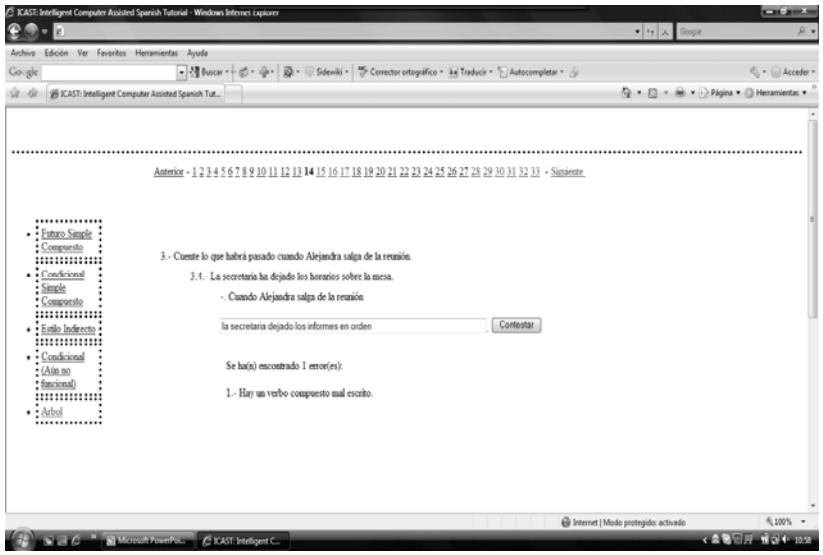

Figura 4: Identificación del tipo de error «Omisión de verbo auxiliar» realizada por el *Parser* ELE-UDEC

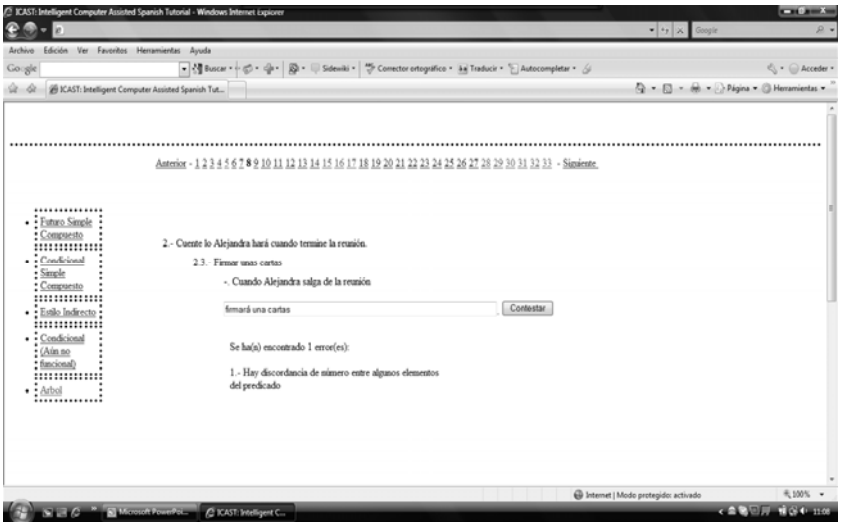

Figura 5: Identificación del tipo de error «Omisión de verbo auxiliar» realizada por el *Parser* ELE-UDEC

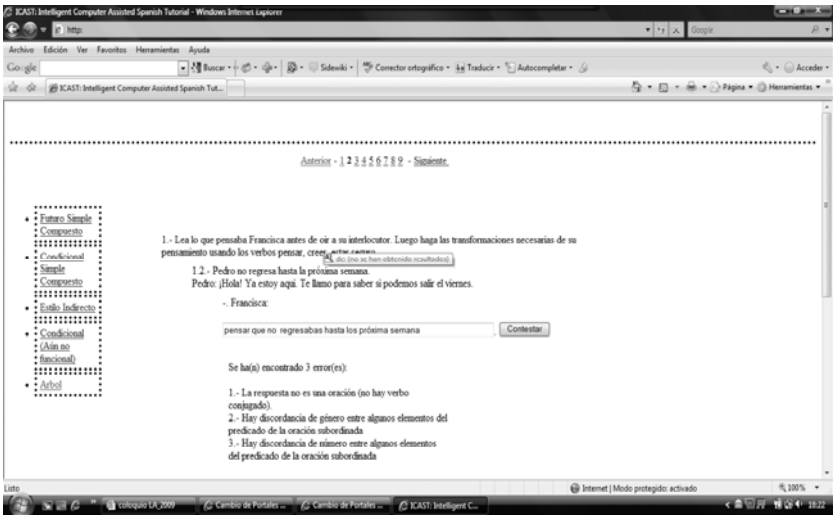

Figura 6: Identificación de Múltiples Errores realizada por el *Parser* ELE-UDEC

Bibliografía

BULL, S. (1997). «Promoting effective learning strategy use in CALL». *Computer Assisted Language Learning*, 10(1):3—39.

CHAPELLE, C.A. (1997). «CALL in the year 2000: Still in search of research paradigms?» *Language Learning and Technology*, 2 (1), 22-34.

DODIGOVIC, M. *Artificial Intelligence in Second Language Learning: Raising Error Awareness*. Clevedon: Multilingual Matters Ltd, 2005.

FERREIRA, A. (2003). *Feedback Strategies for Second Language Teaching with implications for Intelligent Tutorial Systems*. Ph.D. Thesis, University of Edinburgh, Edinburgh,UK.

FERREIRA, A. (2006). *Estrategias Efectivas de Feedback Positivo y Correctivo en Español como Lengua Extranjera*. *Revista Signos*, v.39, n.62, Universidad Católica de Valparaíso.

FERREIRA, A. (2007) «Estrategias efectivas de feedback correctivo para el aprendizaje de lenguas asistido por computadores». *Rev. signos*, 2007, vol.40, no.65, p.521-544.

FERREIRA, A., JOHANNA D. MOORE, Chris MELLISH (2007). «A Study of Feedback Strategies in Foreign Language Classrooms and Tutorials with Implications for Intelligent Computer-Assisted Language Learning Systems». *International Journal of Artificial Intelligence in Education*, 17(4), 389-422.

GRAESSER, A.C., JEON, M., YANG, Y., & CAI, Z. (2007). «Discourse cohesion in text and tutorial dialogue». *Information Design Journal*, 15(3), 199-213. DOC Version

GRAESSER, A.C., PERSON, N., HARTER, D., & TRG (2001). «Teaching tactics and dialog in AutoTutor». *International Journal of Artificial Intelligence in Education*. PDF version

HEIFT, TRUDE and Mathias SCHULZE (2007) *Errors and Intelligence in Computer-Assisted Language Learning. Parsers and Pedagogues.* New York: Routledge.

HOLLAND, V., KAPLAN, J., and SAMS, M. *Intelligent Language Tutors: Theory shaping technology*. Lawrence Erlbaum Associates, 1995.

NAGATA, N. (1995). «A study of consciousness-raising by computer: The effect of metalinguistic feedback on second language learning». *Foreign Language Annals*, 28(3):337–347.

NAGATA, N. (1997). «The effectiveness of computer-assisted metalinguistic instruction: A case study in japanese». *Foreign Language Annals*, 30(2):187–199.

SCHULZE, Mathias (2008) «AI in CALL: Artificially Inflated or Almost Imminent? ». *Calico Journal* 25 (3).

TOMLIN. R. 1995. «Modeling Individual Tutorial Interactions: Theoretical and Empirical Bases of ICALL». In *Intelligent Language Tutors: theory shaping technology*. Lawrence Erlbaum Associates.