CON-TEXT. Un corrector gramatical de bajo nivel

Flora Ramírez Bustamante
Fernando Sánchez León
Laboratorio de Lingüística Informática
Facultad de Filosofía y Letras
Universidad Autónoma de Madrid
E-28049 Madrid (Spain)
{flora;fernando}@maria.lllf.uam.es

Thierry Declerck
DFKI GmbH
(German Research Center
for Artificial Intelligence)
Stuhlsatzenhausweg 3
D-66123 Saarbrücken (Germany)
declerck@dfki.uni-sb.de

Abstract

En este trabajo se presentan los resultados del proyecto CON-TEXT, un corrector gramatical para el español basado en el uso de técnicas de bajo nivel, tales como la segmentación y análisis morfológico de textos. Asimismo, se describen las herramientas de análisis lingüístico del nivel morfológico sobre las que se ha construido este sistema y se presenta la doble arquitectura sobre la que se ha implementado.

1 Antecedentes

El desarrollo de correctores gramaticales automáticos ha estado vinculado a la búsqueda e implementación de técnicas especiales para detectar (y posiblemente corregir) errores para los que es necesario un análisis completo de las oraciones. Estas técnicas se han incorporado a las técnicas tradicionales de procesamiento, ya que una de las características principales de los sistemas de verificación basados en gramáticas o análisis de alto nivel es que han de analizar tanto oraciones correctas como incorrectas. En este sentido, las gramáticas en las que se basan dichos verificadores han de contener tanto la gramática de la competencia de los hablantes como la gramática de la actuación[1], por lo que las técnicas que utilicen han de ser variadas y jerarquizadas, para permitir una detección, diagnóstico y corrección que utilice criterios cognitivos, donde estos estén disponibles, y métodos heurísticos donde los primeros fallen.

Dentro de los formalismos de unificación se han incorporado, principalmente, dos técnicas para la detección y diagnóstico de errores gramaticales: técnicas de relajación de rasgos y técnicas de anticipación de errores. La combinación de ambas técnicas parece ofrecer la aproximación más adecuada para dar cuenta de los errores que interfieren en el análisis sintáctico de las oraciones. Dentro de la creación de correctores gramaticales para el español, el proyecto GramCheck (Ramírez y Sánchez-León, 1996b), (Ramírez y Sánchez-León, 1996c), cuyo objetivo era la implementación de un demostrador de corrección gramatical del español y griego, presenta un uso extensivo de la técnica de relajación para dar cuenta de errores de concordancia intrasintagmática e intersintagmática, problemas en las preposiciones regidas por los núcleos predicativos o la ausencia de preposición *a* ante objetos animados. No obstante, este proyecto también utilizó en determinados casos la técnica de anticipación de errores en contextos locales de error, para los que no se veía imprescindible la implementación de una técnica tan costosa, en términos de tiempo y rendimiento, como la de la relajación[2]. Los resultados de este proyecto, contrastados con los datos que se extrajeron de la tipología de errores que se elaboró, hicieron evidente que, si bien para la detección de errores gramaticales era necesario el uso de técnicas de alto nivel, había una gran variedad de errores, atestiguados en la tipología, cuyo tratamiento mediante estas técnicas resultaba demasiado costoso porque, o bien aparecían en contextos locales y la detección no requería un análisis completo de las oraciones, aun tratándose de

[1] El término actuación se utiliza aquí con un sentido especial que lo relaciona con la producción de mensajes incompletos, con lagunas informativas, incorrecciones e imprecisiones que lo acercan, de algún modo, a lo que definiría como incorrecto un modelo basado en la gramática de la competencia.

[2] Puede encontrarse información precisa sobre las técnicas y estrategias implementadas en este corrector en el sitio `http://www.lllf.uam.es/~flora/projects/GramCheck.html`.

errores de alto nivel, como los de concordancia, o bien constituían errores cuya detección y diagnóstico no necesitaba obligatoriamente de un análisis completo sintáctico (Ramírez y Sánchez-León, 1996a)[3]. Por otra parte, el gran interés que en los últimos años ha suscitado el nivel del preprocesamiento en los sistemas de PLN revelaba la posible combinación de técnicas de bajo y alto nivel como medio para dotar de robustez a estos sistemas[4].

En este contexto se integra el proyecto CON-TEXT[5], cuyo objetivo es el uso de una serie de recursos de análisis de bajo nivel —propios del procesamiento morfosintáctico de corpus (segmentación, análisis morfológico y desambiguación)— como base de conocimiento lingüístico que, unido a la formulación o implementación de un conjunto de reglas lingüísticamente motivadas, demostrara que estas técnicas y conocimiento proporcionan puntos de anclaje fiables y suficientes para la construcción de un prototipo de verificación gramatical que responda a las necesidades y expectativas de los usuarios. Asimismo, otro de los resultados científicos que se esperaba del proyecto se relacionaba con la integración de dichas técnicas y recursos en sistemas de PLN, con el fin de dotarlos de robustez, de forma que tanto sistemas de procesamiento de corpus como sistemas de más alto nivel pudieran beneficiarse de una verificación gramatical y de estilo que normalizara los textos antes de procesarlos, y que, si no eliminara, al menos redujera, las posibilidades de fallo de los sistemas debido al procesamiento de texto erróneo.

Con esta metodología se ha perseguido una simplificación de los métodos computacionales usados tradicionalmente en la verificación gramatical basada en el conocimiento lingüístico, aunque ello haya de realizarse a costa de una limitación en la cobertura del prototipo final. Dicha limitación, en principio, es esperable dada la estrategia localista de la implementación y el nivel de análisis lingüístico, limitado, exclusivamente, al morfológico. La localidad de las reglas impiden detectar errores en contextos no locales. La ambigüedad, por otra parte, conlleva a que no sea posible la detección de ciertos errores en secuencias de formas léxicas ambiguas que bloquean una determinada situación de error. Este compromiso, sin embargo, implica que la precisión del sistema ha de ser superior al de los sistemas comerciales.

Dentro de este proyecto, se han implementado dos versiones de CON-TEXT: una en el entorno `Windows95/NT` y otra en el entorno `UNIX`. La versión para `Windows95/NT` maneja secuencias de errores expresados como autómatas compilados en un solo autómata por composición. La versión `UNIX` trabaja con un conjunto de patrones que aplica secuencialmente hasta la saturación. Aunque existen diferencias en la implementación, ambas versiones resultan equivalentes en cuanto a la cobertura. Aquí nos referiremos únicamente a la arquitectura de la herramienta en versión `UNIX`.

2 Segmentación y Anotación

CON-TEXT es un verificador gramatical basado en la anotación morfológica de textos. La herramienta está implementada en Perl. La estrategia general de detección es la de la anticipación de errores y las técnicas se basan en la búsqueda de cadenas o patrones que satisfacen determinadas condiciones de error. CON-TEXT se compone de un conjunto de reglas locales que describen secuencias erróneas de descripciones morfosintácticas que se contrastan con las descripciones morfosintácticas del texto anotado.

Las herramientas y recursos para la segmentación y anotación utilizados por CON-TEXT proceden de herramientas de dominio público y privadas. En concreto, el segmentador, `mtSeg`[6], procede del proyecto MULTEXT. Las herramientas de análisis morfolólogico (Sánchez-León, 1997) se basan en una morfología flexiva generadora que toma como base un lexicón estructurado en el que conviven información morfo-léxica propia de un lexicón computacional con el aparato de definiciones de un diccionario tradicional. El lexicón, LE^{z}E, incorpora actualmente definiciones del *DRAE* y del *DAL* (*Diccionario ANAYA de la lengua*) y consta de 60.000 lemas. El generador es un `script` en Perl que interpre-

[3] Véase una argumentación parecida en (Oliva, 1997).

[4] Para una discusión sobre este punto, puede consultarse (Ramírez, Sánchez-León y Declerck, 1997)

[5] Este proyecto, de un año de duración, ha sido financiado por la Consejería de Educación y Cultura de la Comunidad Autónoma de Madrid, ref. 05C/002/96.

[6] `mtSeg` puede encontrarse en el sitio: `http://www.lpl.univ-aix.fr/projects/multext/`

ta información paradigmática y aplica una cascada de cambios morfografémicos a los lexemas de entrada de acuerdo con condiciones de concatenación[7]. La idea fundamental con respecto a la verificación ortográfica y gramatical es que el generador es capaz de relajar ciertas reglas morfografémicas y producir un conjunto de formas flexivas que representan errores cognitivos comunes en la morfología flexiva. Estos errores, debido a la distancia gráfica de la forma correcta, no son nunca capturados por los correctores ortográficos. Además de estas formas plenas, el lexicón contiene 1.100 errores cognitivos léxicos no derivados automáticamente.

El análisis morfológico se define, básicamente, como una consulta sobre una tabla *hash* de formas flexivas. Sin embargo, se ha realizado un tratamiento completo de otros fenómenos morfológicos, que van desde la morfología derivativa (tanto prefijos como sufijos, y tanto morfología con cambio de categoría como apreciativa) para las categorías mayores a las formas verbales con enclíticos, lo que permite al sistema analizar muchas más formas textuales. Toda la información relativa a estos componentes se encuentra expresada en un formalismo declarativo, por lo que resulta fácilmente modificable por el potencial usuario.

Junto a esto, el analizador (otro `script` en Perl) es capaz de estimar la información morfosintáctica de palabras desconocidas (aquellas no incluidas en el lexicón y para las que no sea posible un análisis por medio de los componentes de análisis morfológico anteriormente descritos) tomando en consideración información sufijal, tipográfica y el contexto morfosintáctico[8].

3 Descripción del verificador CON-TEXT

CON-TEXT opera extrayendo la información contenida entre los límites oracionales y aplicando las reglas de error en las secuencias que se encuentran dentro de este límite. Por tanto, CON-TEXT no opera con información distribuida en diferentes oraciones o párrafos.

El programa detecta errores de puntuación, errores léxicos cognitivos de concordancia y de secuencias formadas por formas flexivas que presentan casos homógrafos. Todas las reglas son locales. Esta aproximación conlleva algunas limitaciones, como se ha discutido en la sección 1, pero bien se ha considerado suficiente para una primera versión del programa.

Por otra parte, se ha conseguido un alto nivel de precisión en la detección de errores gracias al procesamiento de clases y subclases de ambigüedad. Esto implica que la detección de errores se realiza en contextos no ambiguos, pero, además, en determinados contextos que, siendo ambiguos, contienen una clase o subclase de ambigüedad que no presenta ninguna categoría capaz de bloquear la detección del error.

La detección de un error dispara un mensaje de ayuda para el usuario. Estos mensajes, basados en un diagnóstico heurístico de la secuencia de error detectada, son adecuados a la situación de error en casi todos los casos, y, en aquellos en que no resultan adecuados, marcan tendencias que permiten al usuario realizar su propio diagnóstico y posterior corrección[9].

Los mensajes de error, como las reglas que los disparan, están parametrizados en dos tipos fundamentales: aquellos que se disparan por la detección de un error y aquellos que se disparan por la detección de una secuencia o forma léxica cuya corrección podría normalizar el texto. Los primeros constituyen errores severos; los segundos constituyen debilidades estilísticas que o bien son errores que están muy extendidos en

[7]Esta implementación ha sido fácilmente migrada a un generador morfológico basado en dos niveles y gramáticas de unificación como mmorph, desarrollado como parte de las herramientas de MULTEXT para el procesamiento lingüístico de corpus (Petitpierre y Russell, 1995).

[8]Puede encontrarse información detallada sobre todas estas herramientas en (Sánchez-León, 1997).

[9]Por ejemplo, en la secuencia errónea *el numero*, un humano podría realizar varios diagnósticos que operarían sobre la corrección de uno de los elementos de la secuencia o de ambos, en función de un contexto lingüístico más amplio y profundo que el manejado por CON-TEXT: (a) falta un acento en el sustantivo *número*; esto es, la secuencia correcta es *el número*; (b) falta un acento ortográfico en ambas palabras; esto es, la secuencia correcta es *él numeró*; o (c) falta un acento ortográfico en la forma *el* y, forzado por la concordancia, *numero* debe ser sustituido por la forma de tercera persona *numera*. Las reglas de CON-TEXT detectan que un artículo no puede ir seguido de un verbo, pero son incapaces de diagnosticar la palabra exacta donde se ha producido el error en este contexto.

la lengua, o bien, aunque no son estrictamente errores, se ha considerado que un consejo sobre su forma más correcta podría ayudar al usuario a tomar decisiones sobre la apariencia de su texto. Entre los primeros se encuentran los errores que se mencionan en la sección 4; entre los segundos, se incluyen calcos sintácticos (*la carta a enviar*) y formas flexivas regulares en sustantivos con alguna irregularidad (*déficits*).

El conjunto de reglas está estructurado de tal forma que es posible seleccionar subcomponentes, lo que permite una parametrización en función de las necesidades del usuario. En este sentido, ya se trabaja en la creación de 'perfiles de usuario' (cf. sección 6). Finalmente, las reglas se aplican hasta la saturación, por lo que un mismo contexto puede satisfacer las restricciones de más de una regla.

4 Cobertura

En esta sección, se describe la cobertura del sistema en relación con las reglas que detectan errores severos. Estas incluyen reglas de detección de errores ortográficos sobre la puntuación y reglas de detección de errores morfosintácticos.

Las reglas de detección de errores de puntuación están ancladas en los signos de puntuación existentes. Es decir, la herramienta no proporciona ayuda sobre la colocación de tales signos, pero sí detecta signos mal ubicados en relación con su posición recta o no balanceados si se trata de signos de este tipo. El universo de errores incluye, entre otros, los siguientes: ausencia o adición de espacio después de un signo de puntuación, adición de punto después de un signo de interrogación o admiración, ausencia de punto en determinadas abreviaturas, falta de uno de los signos balanceados de interrogación o admiración, dos o más signos de puntuación incompatibles seguidos, etc.

El segundo tipo de reglas incluye reglas de concordancia que formulan secuencias de descripciones morfosintácticas erróneas con respecto al género y al número dentro de sintagmas nominales. En estas secuencias se permiten determinadas ambigüedades, referidas a macro-clases de palabras (por ejemplo, nominales), con el fin de aumentar la cobertura del sistema. Esta selección asegura que la detección se realiza sobre categorías nominales no ambiguas o ambiguas dentro de esta macroclase, de tal forma que se rechacen secuencias ambiguas inmanejables como *los enfoque*. Dado el alto número de homógrafos en español, esta estrategia mixta permite un adecuado equilibrio entre cobertura y precisión[10].

Asimismo, este módulo contiene reglas para la detección de secuencias erróneas sobre formas flexivas que presentan quasi-homógrafos y que constituyen habitualmente errores cognitivos o de simple descuido (pares léxicos como *se–sé, el–él, cuando–cuándo, mas–más, o–u, y–e, del–de el, tu–tú, a–ha*, etc.)[11]. También se incluye en este grupo la detección de la omisión de una preposición regida en contextos locales (**avisar que, *inferior que, *coetáneo a...*).

La gramática de errores contiene un total de 225 reglas. Las reglas, enunciadas declarativamente, contienen un foco o anclaje y un posible contexto izquierdo y derecho y llevan asociadas un número de error que permite recuperar el mensaje adecuado de la base de datos de errores.

5 Pruebas y resultados

Con el fin de realizar una evaluación de CONTEXT, se han extraído diversos artículos (un total de 44.922 palabras) de diferentes versiones electrónicas de periódicos (*ABC* y *El País*). Además, se han vertido a un texto (12.287 palabras) la colección de oraciones erróneas, atestiguadas en medios de comunicación escrita, a partir de las cuales se compuso nuestra tipo-

[10] Con todo, existen serios problemas y limitaciones en la detección de errores de concordancia, debido a que la herramienta no está basada en ningún tipo de análisis sintagmático. En concreto, dada la aproximación localista, el sistema falla sistemáticamente sobredetectando errores de concordancia, por ejemplo, en predicaciones secundarias: *salieron de la* clase juntos. Estos problemas apuntan hacia la necesidad de niveles de análisis más profundos e información léxica más rica para detectar este tipo de secuencias, que son, por otra parte, menos frecuentes en los textos.

[11] Como puede verse, algunos de estos errores constituyen errores ortográficos que interferirían en el análisis sintáctico de las oraciones. Estos errores no podrían, en ningún caso, ser detectados por herramientas de verificación ortográfica, ya que solamente el procesamiento de la información morfosintáctica de los elementos que rodean al error proporciona un medio para su detección.

logía de errores. Estos textos no han sido utilizados durante la elaboración de las reglas de CON-TEXT.

Los textos se han verificado tanto con CON-TEXT como con otros dos productos comerciales: el corrector gramatical de Word 7 de Microsoft y WordCorrect, un corrector gramatical (y también ortográfico) desarrollado para el español por la empresa DGC - Desarrollo Gramatical Computarizado, de Barcelona[12].

Hemos medido el comportamiento de estas herramientas en términos de cobertura y precisión. La cobertura se calcula dividiendo el el número de errores detectados correctamente por la herramienta por el número real de errores. La precisión se calcula dividiendo el número de errores bien detectados por el número total de errores detectados por la herramienta (Gómez Guinovart, 1996). Para esta prueba, nos hemos limitado a registrar los errores correctamente detectados, dejando a un lado el hecho de que el diagnóstico (esto es, el mensaje de error) pudiera no ser adecuado.

Las tablas 1 y 2 muestran la precisión de la herramienta CON-TEXT, que supera sobradamente la de cualquiera de las dos herramientas comerciales. Hay que hacer notar el deficiente tratamiento de la concordancia y de las secuencias de elementos léxicos dado por estos sistemas. Asimismo, WordCorrect presenta un tratamiento muy pobre de los elementos textuales, tales como abreviaturas, y de los signos de puntuación. Esto provoca que el número de errores detectados aumente considerablemente. Los resultados de estas pruebas ponen de manifiesto que la cobertura de los tres sistemas es, probablemente, similar, aunque la cobertura de CON-TEXT puede decaer en determinados textos donde confluyan de manera especial errores gramaticales que involucren formas léxicas morfosintácticamente ambiguas, ya que existe un compromiso entre cobertura y precisión, de tal forma que se sacrifica la detección de ciertos errores en aras del mantenimiento del alto nivel de precisión.

[12]Los resultados presentados en estas tablas sobre la versión UNIX de CON-TEXT son equivalentes a los obtenidos con la versión Windows95/NT.

6 Arquitectura. Trabajo futuro

Una nueva etapa en la implementación del sistema explota ahora las facilidades de modularización que ofrece perl 5, por lo que los distintos procesos de verificación han sido distribuidos en diferentes paquetes. Estos módulos se llaman y controlan desde una función principal. De esta forma pueden ser fácilmente activados o desactivados. Una simple interfaz permite al usuario activar los distintos procesos desde la línea de comandos.

El mismo proceso de modularización se ha realizado para los mensajes de error, correspondientes a los distintos tipos de verificación textual que se realizan. Los ficheros de mensajes de error son ahora externos al programa, lo que permite a los usuarios modificar el texto de los mensajes existentes sin tener que acceder al programa. Además, el usuario podría tener la oportunidad de añadir nuevos mensajes de error, lo que sería interesante si se le permitiera añadir sus propias reglas de error (esta posibilidad se está considerando para futuras versiones).

En el futuro se prevén nuevas parametrizaciones, como permitir al usuario decidir el tipo de información que desea recibir sobre el patrón de error localizado (el texto completo, las palabras clave —concordancia, tipografía, etc.— o, simplemente, el número de error).

Por otra parte, se están realizando los cambios oportunos para desligar la herramienta de la salida actual de los programas de segmentación y anotación. Sin embargo, en este punto es necesario aún diseñar una interfaz clara entre las reglas de verificación y la salida de diferentes analizadores. Este punto permitiría una evaluación real del programa CON-TEXT basado en diferentes tipos de datos.

Tabla 1: Resultados con los artículos procedentes de *ABC* y *El País*

			CON-TEXT		Word 7		WordCorrect	
# Errores		116	67		157		266	
Tipograf.	No tipograf.	77 | 39	49	18	64	93	117	149
Reales			49	14	48	16	50	25
# Total Errores Reales			63		64		75	
Cobertura			54,31%		55,17%		64,65%	
Precisión			94,02%		40,12%		28,19%	

Tabla 2: Resultados con el corpus de errores

			CON-TEXT		Word 7		WordCorrect	
# Errores		320	124		148		133	
Tipograf.	No tipograf.	133 | 187	85	39	66	82	55	78
Reales			85	35	57	41	45	45
# Total Errores Reales			120		98		90	
Cobertura			37,5%		30,6%		28,1%	
Precisión			96,77%		66,2%		67,6%	

References

Gómez Guinovart, J. 1996. Aportaciones a la Metodología de Evaluación de los Sistemas de Verificación Automática de la Sintaxis In *Procesamiento de Lenguaje Natural*, Revista nº 19, pp. 7–13.

Oliva, K. 1997. Techniques for accelerating a grammar-checker, In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 155–158.

D. Petitpierre and G. Russell. MMORPH - The MULTEXT Morphology Program. MULTEXT deliverable report for the task 2.3.1, ISSCO, University of Geneva, February 1995.

Ramírez Bustamante, F., F. Sánchez-León. 1996. Is linguistic Information enough for grammar checking?, In *Proceedings of the First International Workshop on Controlled Language Applications, CLAW '96*, pp. 216–228.

Ramírez Bustamante, F., F. Sánchez-León. 1996. GramCheck: A Grammar and Style Checker, In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pp. 175–181.

Ramírez Bustamante, F., F. Sánchez-León. 1996. GramCheck. Un corrector gramatical para el español, In *Procesamiento de Lenguaje Natural*, Revista nº 19, pp. 30–37.

Ramírez Bustamante, F., F. Sánchez-León, T. Declerck. 1997. Corrección gramatical y preprocesamiento, In *Procesamiento de Lenguaje Natural*, Revista nº 21, pp. 147–156.

Sánchez-León, F. 1997. Análisis morfosintáctico y desambiguación en castellano, Tesis Doctoral inédita, Universidad Autónoma de Madrid.