Análisis Sintáctico Ascendente con un Algoritmo Evolutivo *

Lourdes Araujo
Dpto. Sistemas Informáticos y Programación.
Universidad Complutense de Madrid. 28040.
lurdes@sip.ucm.es

Resumen: Los métodos de análisis sintáctico clásicos buscan las distintas interpretaciones de una sentencia mediante técnicas de búsqueda completas. Pero el tamaño del espacio de búsqueda crece exponencialmente con la longitud de la sentencia y el tamaño de la gramática, de forma que los métodos de búsqueda exhaustivos pueden ser insuficientes. Sin embargo, existen otras técnicas de búsqueda, como los algoritmos evolutivos, que aunque no garantizan encontrar el valor óptimo, permiten ajustar la calidad de las soluciones obtenidas incrementando el número de puntos explorados. Este trabajo presenta un algoritmo evolutivo para realizar el análisis sintáctico con gramáticas probabilísticas. El algoritmo trabaja con análisis parciales, que los operadores genéticos combinan para extender el análisis a segmentos más largos de sentencia. En el artículo se describen los principales elementos del algoritmo, presentando también los resultados obtenidos para un texto extraído del corpus de Susanne.
Palabras clave: Programación evolutiva, análisis sintáctico parcial, gramáticas probabilísticas

Abstract: Classic parsing methods are based on complete search techniques to find the different interpretations of a sentence. However, the size of the search space increases exponentially with the length of the sentence or text to be parsed and the size of the grammar, so that exhaustive search methods can fail to reach a solution in a reasonable time. Nevertheless, large problems can be solved approximately by some kind of stochastic techniques, which do not guarantee the optimum value, but allow adjusting the probability of error by increasing the number of points explored. This paper presents the implementation of a stochastic parser based on a genetic algorithm. This algorithm produces successive generations of individuals, which are partial parses of a sentence. Genetic operators combine individuals to produce longer parses. The model has been implemented, and the results obtained for a number of sentences extracted form the Susanne corpus are presented.
Keywords: Evolutionary programming, partial parsing, probabilistic grammars

1 Introducción

El análisis sintáctico de una sentencia puede verse como un procedimiento que busca las distintas formas de combinar las reglas gramaticales para encontrar combinaciones que puedan ser la estructura de la sentencia. Un analizador ascendente empieza con la secuencia de clases léxicas de las palabras, y su operación básica es tomar una secuencia de símbolos que encaje con el lado derecho de alguna de las reglas. Un punto clave en el análisis es la necesidad de seleccionar el "mejor" entre la colección de análisis posibles consistentes con la gramática, es decir, de tratar la ambigüedad. El análisis sintáctico estadístico permite tratar esta cuestión. Las gramáticas probabilísticas (Charniak, 1993), que se obtienen añadiendo a las reglas gramaticales su probabilidad, representan un importante método estadístico en la lingüística computacional, que ha permitido avances en áreas como la desambiguación y la corrección de errores. Cuando se utilizan gramáticas probabilísticas, la eficiencia de los analizadores puede mejorarse desarrollando algoritmos que intenten explorar los componentes de alta probabilidad en primer lugar.

Los métodos de análisis clásicos se basan en técnicas de búsqueda completas para encontrar las distintas interpretaciones de una sentencia. Sin embargo, algunos experimen-

* Financiado por el proyecto PR1/03-11588.

tos sobre la forma en que los seres humanos realizan el análisis indican que no realizan una búsqueda completa. Por el contrario, el análisis humano parece estar más cercano a un proceso heurístico con algún componente aleatorio. Esto sugiere explorar métodos de búsqueda alternativos.

Los Algoritmos Evolutivos (AEs) representan un método de búsqueda alternativo. Un AE produce sucesivas generaciones de una población de individuos, de los que calcula una medida de su calidad o "aptitud", de acuerdo con la que selecciona la población de la siguiente generación. Ya han sido aplicados a algunos problemas de procesamiento de lenguaje natural (Kool, 2000), como la traducción de preguntas (M. Davis, 1996), la inferencia de gramáticas de contexto libre (Smith y Witten, 1995), el etiquetado léxico (Araujo, 2002b) y el análisis sintáctico (Araujo, 2002a). Este último sistema genera y combina aleatoriamente árboles de análisis para la sentencia completa. La función de aptitud de los individuos es la encargada de asignar bajas probabilidades de sobrevivir a aquellos árboles que no se correspondan con un análisis correcto respecto a la gramática. El sistema funciona correctamente sobre un conjunto de sentencias simples, pero los tamaños de población que requiere para analizar sentencias reales con gramáticas reales, como las extraídas de un corpus, es demasiado grande para ser viable.

Este trabajo presenta un algoritmo evolutivo para realizar el análisis sintáctico ascendente con gramáticas de contexto libre (GCL) probabilísticas. Los individuos son análisis parciales, es decir, análisis de segmentos de la sentencia. Los operadores genéticos combinan estos análisis parciales para ir extendiendo el análisis a segmentos más largos de sentencia. La naturaleza de los algoritmos evolutivos, que favorece la exploración de nuevas áreas del espacio de búsqueda, ayuda a alcanzar el resultado correcto, incluso si éste incluye reglas de baja probabilidad.

El resto del artículo se organiza de la siguiente forma: La sección 2 describe el analizador evolutivo, presentando los elementos principales de su diseño. La sección 3 presenta y analiza los resultados experimentales, y finalmente, la sección 4 resume las conclusiones del trabajo.

2 El Analizador Sintáctico Evolutivo

Los datos de entrada al analizador, aparte de los parámetros propios del algoritmo evolutivo (tamaño de población, número de generaciones, etc), son el texto con las sentencias a analizar, un diccionario del que se obtienen las etiquetas léxicas de las palabras y su frecuencia, y la gramática de contexto libre probabilística.

Consideremos los principales elementos del algoritmo evolutivo.

2.1 Representación de los Individuos

Los individuos en este sistema son análisis de segmentos de la sentencia, es decir, árboles obtenidos aplicando la gramática probabilística a una secuencia de palabras de la sentencia. Cada individuo tiene asignada una categoría sintáctica que es el lado izquierdo de la regla de la cima del árbol de análisis. La figura 1 muestra algunos individuos correspondientes al análisis de la sentencia *The new promotion manager has been employed by the company since January +, 1946 +, as a commercial artist in the advertising department +.*, extraída del corpus de Susanne. Podemos observar que hay individuos compuestos de una única palabra, como el 1. En estos casos la categoría del individuo es la categoría léxica elegida para la palabra (una palabra puede pertenecer a más de una categoría). Así la categoría del individuo 1 es *AT1*. Otros individuos como el 4, cuya categoría es *P*, son árboles de análisis obtenidos aplicando una serie de reglas gramaticales.

2.1.1 Población Inicial

El primer paso del análisis de una sentencia dada es encontrar las etiquetas léxicas válidas de cada palabra. Estas se obtienen de un diccionario, junto con sus frecuencias.

Puesto que nuestros análisis se construyen de forma ascendente, la población inicial se compone de individuos que son árboles compuestos únicamente por la categoría léxica asignada a una palabra. El sistema genera un individuo para cada una de las categorías léxicas de la palabra. Para mejorar la eficiencia, la población inicial también incluye los individuos que se obtienen aplicando aquellas reglas gramaticales cuyo lado derecho está compuesto exclusivamente de categorías léxicas. El individuo 5 de la figura 1

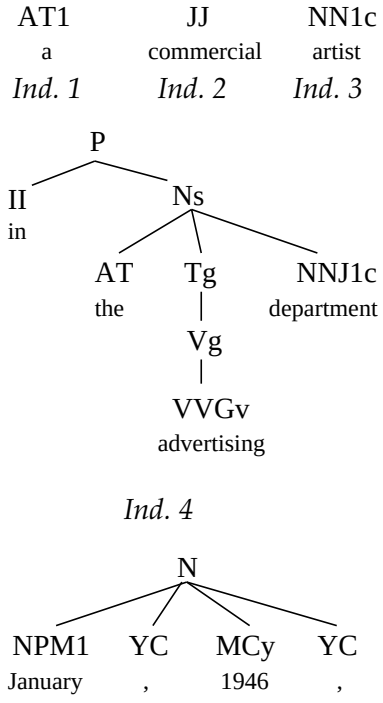

Figura 1: Ejemplos de individuos en el análisis de la sentencia *The new promotion manager has been employed by the company since January +, 1946 +, as a commercial artist in the advertising department +.*

es un ejemplo.

2.2 Aptitud: Evaluación de los Individuos

La evaluación de los individuos es crucial puesto que las oportunidades de sobrevivir de un individuo dependen de su aptitud. La adaptación de los individuos se revisa después de cada generación. La función de aptitud es una medida de la probabilidad del análisis, que se calcula como la media de la probabilidad de las reglas gramaticales usadas en su construcción:

$$aptitud = \frac{\sum_{\forall s_i \in T} prob(s_i)}{nn(T)}$$

donde T es el árbol a evaluar, s_i cada uno de sus nodos y $nn(T)$ el número de nodos. Para las categorías léxicas, la probabilidad es la frecuencia relativa de la etiqueta elegida.

2.3 Operadores Genéticos

Los individuos nuevos que aparecen en una generación se crean mediante dos operadores genéticos: el *cruce* y el *corte*. El operador de cruce combina un árbol de análisis con otros que ya se encuentren en la población para satisfacer el lado derecho de una regla gramatical. El operador de corte crea un nuevo análisis seleccionando aleatoriamente un subárbol del individuo al que se aplica. Los porcentajes de aplicación de estos operadores en cada generación son parámetros de entrada, a los que es muy sensible la eficiencia del sistema.

Un operador genético clásico que no se usa en este sistema es la *mutación*. Este operador realiza aleatoriamente cambios puntuales en la estructura de un individuo, con el objetivo de introducir más variabilidad en la población. Una de las premisas de nuestro sistema es que sólo permite individuos válidos. Por lo tanto, si la mutación se implementa como un cambio aleatorio en un nodo o en un subárbol del individuo, la probabilidad de que produzca un individuo válido es extremadamente baja. Una posibilidad sería que la mutación sustituyera un subárbol por otro individuo de la población que analizara exactamente la misma secuencia de palabras y con el mismo símbolo sintáctico que el sustituido. Pero este mismo efecto se obtiene con el operador de cruce, por lo que la mutación no se ha implementado explícitamente.

Los nuevos individuos que producen los operadores genéticos en cada generación se añaden a la población anterior, que de esta forma va creciendo. El proceso de selección es el encargado de reducir el tamaño de población hasta el valor especificado en los parámetros de entrada al algoritmo. La selección favorece la supervivencia de los individuos de mayor aptitud, pero también tiene en cuenta otros factores que aseguran la presencia en la población de árboles de análisis que contiene palabras que pueden necesitarse en generaciones posteriores. La velocidad de convergencia del algoritmo se acelera mediante la introducción de elitismo, es decir, se asegura la supervivencia de una pequeña porción de la población compuesta por los mejores individuos. El mecanismo evolutivo continúa hasta un número máximo de generaciones establecido o hasta alcanzar la condición de convergencia. Esta condición requiere que el mejor individuo de la población sea un análisis de la sentencia completa, y que no cambie durante un número específico de generaciones.

2.3.1 Reproducción

El operador de cruce produce un nuevo individuo combinando algunos individuos de la población anterior. En primer lugar se selecciona un individuo de la población. En función de este individuo se selecciona una regla gramatical cuyo lado derecho comience con el símbolo sintáctico de este individuo. A continuación se buscan en la población individuos cuyos símbolos sintácticos se correspondan con el resto del lado derecho de la regla. Finalmente, se construyen nuevos individuos cuyo símbolo sintáctico es el lado izquierdo de la regla y cuyo árbol está formado por distintas combinaciones de los individuos seleccionados para completar la regla.

Supongamos que se selecciona el individuo *1* de la Figura 1. Entonces el proceso de cruce sería el siguiente:

- Se identifica la categoría sintáctica del árbol seleccionado, que es *AT1*.
- Se seleccionan aquellas reglas gramaticales cuyo lado derecho empieza con esa categoría sintáctica. Algunos ejemplos de tales reglas en la gramática que se usa en este trabajo son:

  Ns → AT1 JJ NN1c P
  Ns → AT1 JJ NN1n P
  Ns → AT1 JJ Tg NN1c P
  Ns → AT1 NN1c Po
  Ns → AT1 NN1c YC Nns YC MCn

  Consideramos el proceso para la primera de ellas.
- Se buscan en la población individuos de cada categoría del lado derecho de la regla, excepto la primera, comprobando que la secuencia de palabras que analizan estos individuos es la continuación de la de los anteriores individuos seleccionados. En el ejemplo buscamos un individuo de categoría *JJ* cuya secuencia de palabras empiece por *commercial* (la siguiente a la secuencia del individuo *1*), y seleccionamos el individuo *2*. A continuación se busca un individuo de categoría *NN1c* que comience por *artist*, como el individuo *3*. Finalmente, se busca un individuo de categoría *P* que comience con *in*, seleccionando el individuo *4*. Si hubiera otras elecciones posibles se considerarían todas ellas.
- Una vez seleccionados los individuos se construye un nuevo árbol que tiene como raíz la categoría sintáctica de la regla (*Ns*) y como subárboles los árboles seleccionados en el paso previo. El individuo resultante aparece en la Figura 2.
- Se añade el nuevo individuo a la población.

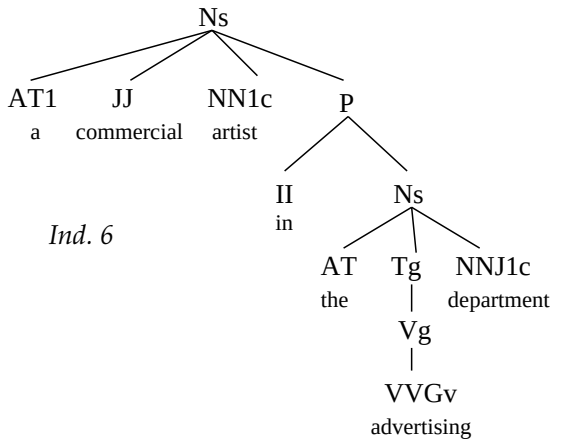

Figura 2: Ejemplo de individuo resultante del operador de cruce.

Con este esquema, puede ocurrir que el cruce de un individuo no produzca ningún descendiente o que produzca más de uno. En este último caso, todos ellos se añaden a la población. El proceso de selección posterior se encarga de reducir después el tamaño de la población al valor especificado.

Si este esquema se aplicara de forma aislada, el tamaño medio de los individuos se incrementaría en cada generación. Aunque esto parece ventajoso porque en última instancia nos interesan individuos que analizan la sentencia completa, también puede originar ciertos problemas. Si el proceso de selección elimina individuos pequeños que sólo pueden combinarse con otros que se producirán en generaciones posteriores, nunca se producirían los análisis correspondientes a estas combinaciones. Esta situación se evita introduciendo algunas condiciones especiales en el proceso de selección, y mediante el operador de *corte*.

2.3.2 El operador de Corte

El porcentaje de aplicación del operador de corte se incrementa con la longitud de la secuencia de palabras analizada por los individuos. Así, la aplicación del operador de corte

a un individuo depende de dos parámetros, *PC* (porcentaje de corte) y *UC* (umbral de corte). *PC* es el porcentaje de aplicación del operador, mientras *UC* es el número mínimo de palabras que se exigen para permitir la aplicación del *corte*, y se especifica como un porcentaje de la longitud de la sentencia que se está analizando. La Figura 3 muestra un esquema del operador. Para cada individuo (*ind*) de la población se comprueba si se cumplen las condiciones de aplicación del *corte*. Si es así, se selecciona aleatoriamente un subárbol del individuo (*SubárbolAlea*), con el que se construye un nuevo individuo que se añade a la población.

```
función Corte(Pob, PC, UC)
    para cada ind en Pob hacer{
        si número_palabras(ind) > UC {
            prob = random(100);
            si (prob < PC) {
                nuevo = SubárbolAlea(ind);
                Añadir(Pob, nuevo);
            }
    }
fin
```

Figura 3: Esquema del operador de *corte*. *Pob* representa la población.

2.4 El Proceso de Selección

Habitualmente, la selección reemplaza los individuos menos adaptados de la población por individuos generados mediante los operadores genéticos. Sin embargo, hay dos cuestiones que hacen que en nuestro caso la selección sea algo diferente. En primer lugar, todos los individuos que generan los operadores genéticos se han incluido en la población, que por tanto necesita reducirse para mantener un tamaño adecuado. Y en segundo lugar, si la aptitud fuera el único criterio para seleccionar los individuos a eliminar, podrían desaparecer individuos que sean los únicos cuyo análisis incluye una determinada palabra de la sentencia, haciendo entonces imposible generar un análisis de la sentencia completa en generaciones posteriores. Por lo tanto, nuestro proceso de selección reduce el tamaño de la población eliminando individuos de acuerdo con su aptitud siempre que todas las palabras de la secuencia que analizan estén presentes en al menos otro individuo de la población.

3 *Relación con un Analizador Sintáctico tipo* chart

El analizador evolutivo (AE) puede verse como una implementación de un analizador ascendente de tipo *chart*. En este tipo de analizador se usa una estructura de datos denominada *chart* para almacenar los resultados parciales de los encajes de reglas ya realizados, evitando así que el analizador intente repetidamente los mismos encajes. La estructura *chart* recoge todos los componentes que se van derivando en el proceso, así como las reglas para las que se ha encontrado un encaje parcial, pendiente de completar. Estas últimas se denominan *arcos activos*. La operación básica de un analizador de tipo *chart* es combinar un arco activo con un componente completo. El resultado es o bien un nuevo componente completo o un nuevo arco activo que extiende el arco original. Los componentes completos se almacenan en una lista denominada *agenda* hasta que se sacan para añadirlos a la estructura *chart*. Este proceso se denomina *algoritmo de extensión de arcos*. Cuando se utilizan gramáticas probabilísticas, estos analizadores pueden modificarse fácilmente para considerar los componentes más probables en primer lugar.

La población del AE está compuesta de análisis parciales de distintas secuencias de palabras de la sentencia, que se van combinando para producir nuevos análisis de secuencias más largas. Esta población de análisis parciales representa la *agenda* de un analizador de tipo *chart*, que almacena los componentes completos. El algoritmo de extensión de arcos está representado por el operador de cruce del AE, que combina análisis parciales hasta completar las categorías que requiere el lado derecho de una regla, produciendo un nuevo componente completo de la clase del lado izquierdo de la regla. La principal diferencia entre ambos tipos de analizador está en la selección de los análisis parciales que se van a combinar, ya que mientras en un analizador *chart* probabilístico siempre se toma el más probable, en el AE esta selección es aleatoria, aunque sesgada hacia los más probables. Aunque existen estas correspondencias entre los elementos del AE y los de un analizador de tipo *chart*, la componente aleatoria que introduce el AE permite explorar distintas zonas del espacio de búsqueda, sin realizar un recorrido exhaustivo.

4 *Resultados Experimentales*

El algoritmo se han implementado en lenguaje C++ sobre un procesador Pentium III. El analizador se ha evaluado sobre un conjunto de sentencias extraídas del corpus de Susanne (Sampson, 1995), una base de datos de sentencias en inglés anotadas sintácticamente. El conjunto de etiquetas léxicas utilizado es una extensión de las etiquetas del corpus de Lancaster y está compuesto de 357 etiquetas. El conjunto de etiquetas sintácticas recoge más de 120 clases.

La gramática utilizada para el análisis se ha obtenido a partir de los análisis de las sentencias del corpus de Susanne, sin considerar las sentencias con referencias externas o de traza. A cada regla gramatical se le asigna una probabilidad calculada como la frecuencia relativa de la regla respecto de la de otras reglas con el mismo lado izquierdo.

Para la evaluación de los análisis obtenidos se han utilizado las medidas de *PARSEVAL precisión*, *cobertura* (*recall*) y *exactitud* (*accuracy*), las más usuales en la evaluación del análisis sintáctico. La *precisión* mide el número de paréntesis del análisis a evaluar que se corresponden con aquellos del árbol correcto, la *cobertura* mide cuantos paréntesis del árbol correcto están en el análisis y la *exactitud* es el porcentaje de paréntesis del análisis a evaluar que no cruzan paréntesis del análisis correcto.

Como algunos autores han señalado (Charniak, 1996), existen varias razones por las que un analizador probabilístico puede producir un análisis incorrecto de una sentencia. Una de ellas es que las reglas necesarias para el análisis no estén en la gramática. Cuando se usa el corpus de Susanne este es un serio problema, debido a que el gran tamaño de los conjuntos de etiquetas léxicas y sintácticas de este corpus hacen que las reglas gramaticales que utiliza sean muy específicas de cada caso. Sin embargo, puesto que lo que nos interesa aquí es evaluar el analizador, este problema puede obviarse aplicando el analizador a sentencias incluidas en el corpus de entrenamiento.

El analizador se ha aplicado a una serie de sentencias, pertenecientes al conjunto de entrenamiento, que componen un texto de 520 palabras, siendo 30 la longitud media de las sentencias. Para comparar el algoritmo evolutivo con un analizador clásico, se ha implementado un analizador tipo *chart* que selecciona en primer lugar los componentes más probables. La Tabla 1 muestra los resultados (el mejor de diez ejecuciones con una desviación estándar menor de 0.2) obtenidos para gramáticas de distintos tamaños, extraídas a partir de distintos conjuntos de entrenamiento del corpus (un conjunto de entrenamiento de 559 palabras para la gramática de 225 reglas, uno de 1652 palabras para la de 466 reglas y uno de 2197 palabras para la de 795 reglas). Podemos observar que los resultados del analizador evolutivo mejoran los del clásico. Aunque estos resultados empeoran ligeramente al aumentar el tamaño de la gramática, pueden mejorarse de nuevo ajustando los parámetros del algoritmo evolutivo.

		Prec.	Cob.	Exac.	E.L.
225r	AC	99.23	99.23	98.20	100
	AE	100	100	100	100
446r	AC	99.23	99.23	98.20	100
	AE	99.01	99.01	99.01	100
795r	AC	99.23	99.23	98.20	100
	AE	97.48	94.86	97.42	99.61

Tabla 1: Resultados de precisión, cobertura y exactitud obtenidos para distintos tamaños de gramática (225, 446 y 795 reglas) con un analizador *chart* probabilístico (AC) y con el analizador evolutivo (AE). La última fila (E.L.) muestra la precisión del etiquetado léxico. El algoritmo evolutivo usa una población de 100 individuos, un número máximo de 40 generaciones, un porcentaje de cruce del 40% y un porcentaje de corte del 30%.

4.1 Estudio de los Parámetros del AE

Se han realizado algunos experimentos para determinar los valores más apropiados para los parámetros del algoritmo evolutivo en el caso de la primera gramática. La Tabla 2 muestra los resultados obtenidos para distintos tamaños de la población cuando se fijan los restantes parámetros. Una población de 100 individuos es el tamaño mínimo requerido para conseguir el análisis completo de todas las sentencias del texto de prueba en 40 generaciones. Con este número de generaciones, agrandar la población empeora los resultados.

La Tabla 3 muestra como mejoran los re-

TP	Precisión	Cobertura	Exactitud
100	100	100	100
150	98.89	98.89	97.42
200	98.89	98.89	97.42

Tabla 2: Resultados obtenidos para distintos tamaños de población (TP), con un número máximo de generaciones de 40, un porcentaje de cruce del 40%, un porcentaje de corte de 30%, y un valor umbral de aplicación del corte de |s| / 3, donde |s| es la longitud de la sentencia.

NG	Precisión	Cobertura	Exactitud
30	98.12	98.12	95.63
35	99.23	99.23	98.20
40	100	100	100
45	100	100	100

Tabla 3: Resultados obtenidos para distintos números máximos de generaciones (NG), con una población de 100 individuos, un porcentaje de cruce de 40%, un porcentaje de corte de 30%, y un umbral de corte de |s| / 3.

sultados con el número de máximo de generaciones que se permite evolucionar al algoritmo. Un número de generaciones menor de 30 no permite obtener el análisis completo de todas las sentencias con los valores elegidos para los restantes parámetros.

Otros parámetros que se han investigado son los porcentajes de aplicación de los operadores genéticos. La Tabla 4 muestra como mejoran los resultados al aumentar el porcentaje de cruce hasta cierto valor (40). Más allá de este valor los resultados empeoran ligeramente, ya que porcentajes más altos de cruce requieren mayores tamaños de población, como muestra la última fila de la tabla. La Tabla 5 muestra que los resultados también mejoran con el porcentaje de aplicación del operador de corte.

También se ha investigado el valor umbral de la longitud de la secuencia de palabras analizada por individuo para permitir la aplicación del operador de corte. Los mejores resultados se obtienen cuando el corte sólo puede aplicarse cuando ésta longitud es mayor que la tercera parte de la longitud de la sentencia.

PX	Precisión	Cobertura	Exactitud
25	97.14	97.14	94.65
30	97.91	97.91	96.44
40	100	100	100
50(a)	99.23	99.23	98.20
50(b)	100	100	100

Tabla 4: Resultados obtenidos para distintos porcentajes de cruce (PX), con un tamaño de población de 100 individuos (excepto en la última fila, que corresponde a 200 individuos), un número máximo de generaciones de 40, un porcentaje de corte de %30 y un umbral de aplicación del corte de |s| / 3.

PC	Precisión	Cobertura	Exactitud
5	97.19	97.19	95.63
10	98.12	98.12	96.44
20	98.12	98.12	96.44
30	100	100	100

Tabla 5: Resultados obtenidos para distintos porcentajes del operador de corte (PC), con un tamaño de población de 100 individuos, un número máximo de generaciones de 40, un porcentaje de cruce de 40% y un umbral de corte de |s| / 3.

5 Conclusiones

Este trabajo describe la implementación de un analizador ascendente probabilístico mediante un algoritmo evolutivo. Dada una sentencia y una gramática, este algoritmo trabaja con una población de análisis parciales de la sentencia. El algoritmo permite un tratamiento estadístico del proceso de análisis, a la vez que introduce una componente aleatoria. De esta forma, ajustando adecuadamente los parámetros, se muestre-

Umbral	Precisión	Recall	Accuracy
|s|	98.12	98.12	95.63
|s|/1.5	98.89	98.89	97.42
|s|/2.0	98.89	98.89	97.42
|s|/2.5	99.01	99.01	99.01
|s|/3.0	100	100	100

Tabla 6: Resultados obtenidos para distintos umbrales de la longitud de la secuencia de palabras analizada para la aplicación del corte.

an distintas zonas del espacio de búsqueda sin recorrerlo exhaustivamente, lo que permite obtener soluciones de alta calidad en un tiempo razonable.

La gramáticas y sentencias usadas para evaluar el sistema se han extraído del corpus de Susanne y se han calculado medidas de *precision*, *cobertura* y *exactitud*. Los resultados obtenidos en estos experimentos mejoran los de un analizador tipo *chart* probabilístico, lo que indica que el paradigma de la programación evolutiva es adecuado para abordar este problema, permitiendo explotar la componente indeterminista del lenguaje natural. También se han realizado una serie de experimentos para ajustar los parámetros del algoritmo evolutivo. Los resultados indican que los valores de estos parámetros están interrelacionados, y así por ejemplo porcentajes más altos de aplicación de los operadores genéticos de cruce y corte requieren un mayor tamaño de población para alcanzar rápidamente un análisis completo de todas las sentencias.

Otra conclusión de este trabajo es que el corpus de Susanne no es apropiado para el tratamiento estadístico del lenguaje, ya que el conjunto de etiquetas léxicas y sintácticas que usa son demasiado grandes para obtener estadísticas significativas, y para dar lugar a una gramática suficientemente general. Por lo tanto, en el futuro se evaluará este analizador sobre otro corpus lingüístico más adecuado. Así mismo, se investigará la aplicación de otras funciones de aptitud y operadores genéticos.

Bibliografía

Araujo, L. 2002a. A parallel evolutionary algorithm for stochastic natural language parsing. En *Proc. of the Int. Conf. Parallel Problem Solving from Nature (PPSNVII)*.

Araujo, L. 2002b. Part-of-speech tagging with evolutionary algorithms. En *Proc. of the Int. Conf. on Intelligent Text Processing and Computational Linguistics (CICLing-2002), Lecture Notes in Computer Science 2276*, páginas 230–239. Springer-Verlag.

Charniak, E. 1993. *Statistical Language Learning*. MIT press.

Charniak, E. 1996. Tree-bank grammars. En *Proc. of the Thirteenth National Conference on Artificial Intelligence*, volumen 2, páginas 1031–1036. AAAI Press / MIT Press.

Kool, Anne. 2000. Literature survey.

M. Davis, T. Dunning. 1996. Query translation using evolutionary programming for multilingual information retrieval II. En *Proc. of the Fifth Annual Conf. on Evolutionary Programming*. Evolutionary Programming Society.

Sampson, G. 1995. *English for the Computer*. Clarendon Press, Oxford.

Smith, T.C. y I.H. Witten. 1995. A genetic algorithm for the induction of natural language grammars. En *Proc. IJCAI-95 Workshop on New Approaches to Learning Natural Language*, páginas 17–24, Montreal, Canada.