Boletín de Estadística e Investigación Operativa
Vol. 34, No. 2, Julio 2018, pp. 159-175

Opiniones sobre la profesión

Natural Language Parsing: Progress and Challenges

Carlos Gómez-Rodríguez
Universidade da Coruña
FASTPARSE Lab, LyS Research Group
Departamento de Computación
Facultade de Informática, Elviña
15071 A Coruña, Spain
✉carlos.gomez@udc.es

Resumen

Natural language parsing is the task of automatically obtaining the syntactic structure of sentences written in a human language. Parsing is a crucial step for language processing systems that need to extract meaning from text or speech, and thus a key technology of artificial intelligence. This article presents an outline of the current state of the art in this field, as well as reflections on the main challenges that, in the author's opinion, it is currently facing: limitations in accuracy on especially difficult languages and domains, psycholinguistic adequacy, and speed.

Keywords: Natural language parsing, syntax, artificial intelligence.
AMS Subject classifications: 68T50, 91F20.

1. Análisis sintáctico del lenguaje natural

El procesamiento del lenguaje natural es la rama de conocimiento que investiga la manera de que las máquinas puedan comunicarse con las personas utilizando lenguajes humanos. Como tal, es un campo interdisciplinar que se puede enmarcar tanto en la inteligencia artificial como en la lingüística computacional. Dentro de este campo, el análisis sintáctico del lenguaje natural es la tarea consistente en obtener, de forma automática mediante un programa de ordenador, la estructura interna de una oración.

Aunque la investigación en análisis sintáctico del lenguaje natural tiene varias décadas de historia, sólo recientemente ha pasado de ser un campo de investigación prometedor a experimentar un uso generalizado en distintas aplicaciones de inteligencia artificial, como la traducción automática [47, 50], reconocimiento de implicaciones textuales [44], aprendizaje para agentes inteligentes en juegos

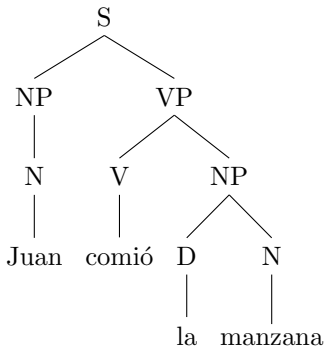

Figure 1: árbol de constituyentes para una oración en español. La etiqueta de cada nodo interno indica el tipo del constituyente formado por las palabras que descienden de dicho nodo. Por ejemplo, "la manzana" es una frase nominal (NP), mientras que "comió la manzana" es una frase verbal (VP).

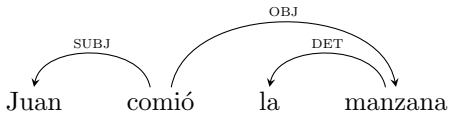

Figure 2: árbol de dependencias para una oración en español. Cada dependencia se representa como una flecha, que va de una palabra (padre) a otra (dependiente). Por ejemplo, la palabra "Juan" depende de "comió", y su tipo de dependencia es SUBJ (sujeto).

[2] o análisis de sentimiento [25, 48]; y convertirse en un componente clave de los sistemas desplegados por las grandes compañías de servicios informáticos, como IBM [38] o Google[1].

La estructura de las oraciones se puede describir mediante distintas representaciones sintácticas, dependiendo de la teoría lingüística que uno siga. En el *análisis sintáctico de constituyentes*, o *análisis sintáctico de estructura de frase*, la estructura de una oración se representa mediante un árbol que la divide en unidades más pequeñas llamadas *constituyentes*, que a su vez se dividen en otros constituyentes más pequeños hasta llegar al nivel de palabras individuales, como en la Figura 1. La otra representación predominante es la del *análisis sintáctico de dependencias*, en la que la estructura de la oración se expresa mediante un árbol o bosque compuesto de relaciones binarias dirigidas entre palabras, llamadas *dependencias*, cada una de las cuales enlaza un *padre* con un *dependiente*, como en la Figura 2.

Aunque la adecuación lingüística de cada una de estas representaciones es

[1] Google Cloud Natural Language: `https://cloud.google.com/natural-language/`

un tema polémico entre los sintacticistas – véase por ejemplo [7, 24] –, los que estamos más sesgados hacia la ingeniería tendemos a centrarnos en "cualquier cosa mientras funcione". Con "funcionar", en este caso, nos referimos a ser capaz de proporcionar una representación de las oraciones que sea útil a las aplicaciones que la utilicen para extraer información sobre el significado del texto, y a hacerlo de manera tan precisa y eficiente como sea posible. Desde este punto de vista, ambas representaciones tienen sus méritos. La sintaxis de dependencias es en la actualidad la aproximación predominante en lingüística computacional y procesamiento del lenguaje natural, dado que se puede decir que es más simple (la representación resultante no tiene más nodos que las propias palabras de entrada), haciendo posible la creación de algoritmos más eficientes, y la salida proporciona una representación sencilla y transparente del significado de la oración (el árbol de la Figura 2 nos dice qué acción sucede en la oración, quién la lleva a cabo y quién la recibe: Juan, el sujeto, se comió la manzana, el objeto). Sin embargo, tampoco procede ignorar la sintaxis de constituyentes: además de proporcionar información distinta y complementaria a la representada por las dependencias [26], existe la paradoja de que algunos de los mejores analizadores sintácticos de dependencias son en realidad analizadores de constituyentes [34], donde primero se obtiene un árbol de constituyentes para después transformarlo en dependencias a través de reglas heurísticas.

Independientemente de la representación que se use, la principal dificultad de conseguir que las máquinas analicen correctamente los lenguajes humanos se localiza en una de las características fundamentales de dichos lenguajes: su *ambigüedad*. Una oración dada puede tener diferentes significados, todos ellos sintácticamente correctos. Por ejemplo, en la oración "Juan vio un hombre con un telescopio", ¿usó Juan un telescopio para ver al hombre (como se refleja en el árbol de constituyentes de la Figura 3, donde la frase preposicional que menciona el telescopio es independiente de la frase nominal que hace referencia al hombre)? ¿O era el hombre el que llevaba un telescopio (como en el árbol de constituyentes de la Figura 4, donde la frase preposicional está anexa a la frase nominal)? Un humano podría desambiguar la oración a partir del contexto (o preguntar, si no estuviera claro). En el caso de una máquina, recurrimos a modelos probabilísticos o de aprendizaje automático para intentar resolver las ambigüedades, con la ayuda de datos etiquetados. Profundizaremos en esto en la siguiente sección, que presenta brevemente diferentes aproximaciones populares para el análisis sintáctico.

2. Modelos de análisis sintáctico y estado del arte

Las primeras aproximaciones al análisis sintáctico del lenguaje natural que se pueden considerar exitosas, en el sentido de proporcionar análisis útiles para oraciones reales, fueron analizadores estadísticos de constituyentes basados en

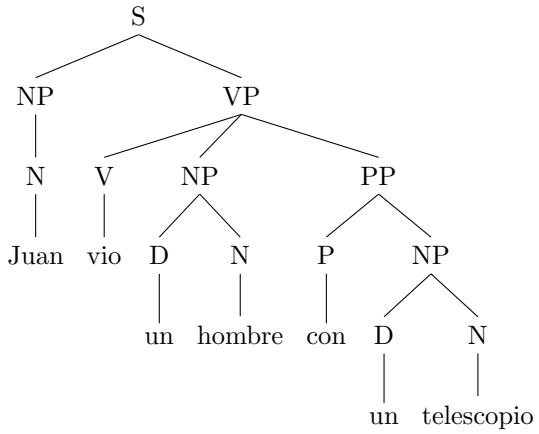

Figure 3: En esta interpretación de la oración, Juan usó un telescopio para ver a un hombre.

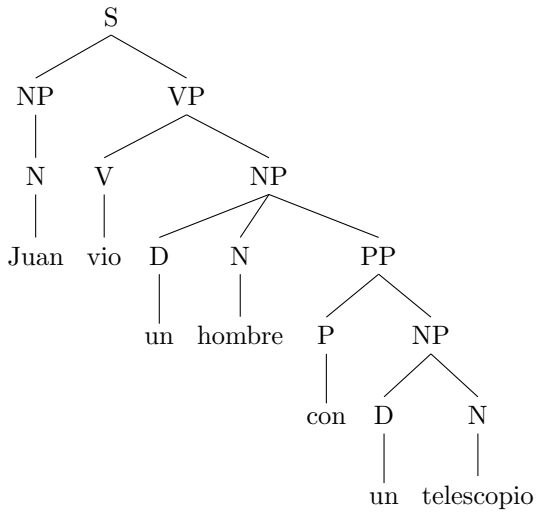

Figure 4: En esta interpretación de la oración, Juan vio a un hombre que llevaba un telescopio.

gramáticas independientes del contexto probabilísticas.

Informalmente hablando, una gramática independiente del contexto es una descripción de la sintaxis de un lenguaje dado por medio de una serie de reglas que describen las estructuras que puedan aparecer en un árbol de constituyentes. Por ejemplo, para describir cómo una frase verbal (VP) se puede dividir en constituyentes más pequeños, podríamos tener una regla VP → V NP PP (que dice que un constituyente de tipo VP puede tener como hijos, de izquierda a derecha, los constituyentes V, NP y PP, como en la figura 3) y una regla VP → V NP (afirmando que un constituyente VP puede tener a V y NP como hijos, como en la Figura 4). Si escribimos una gramática independiente del contexto completa para describir la sintaxis de un idioma dado, como puede ser el inglés, podremos utilizar un algoritmo de programación dinámica, como el algoritmo CKY [28] o el algoritmo de Earley [9], para recuperar los árboles de constituyentes que sean compatibles con una oración dada (véase [17] para una revisión y formalización de distintos algoritmos de programación dinámica de este tipo, sus características y las relaciones entre ellos).

Sin embargo, analizar con gramáticas independientes del contexto puras tiene dos limitaciones importantes cuando damos el salto desde ejemplos "de juguete" a oraciones reales: en primer lugar, construir una gramática completa de un idioma a mano es muy difícil (o incluso imposible, dado que el idioma está en constante evolución); y en segundo lugar, necesitamos tratar la ambigüedad: en muchos casos pueden existir varios árboles compatibles con la gramática (como los de las Figuras 3 y 4) y para la mayoría de las aplicaciones, es necesario elegir el mejor de ellos.

Ambos problemas pueden ser mitigados haciendo que las gramáticas independientes del contexto sean *probabilísticas*. Para ello, a cada regla de la forma $\mathrm{X} \to \alpha$ (donde α es una cadena de símbolos) le asociamos una probabilidad $P(\mathrm{X} \to \alpha)$, de tal modo que

$$\sum_{\alpha} P(\mathrm{X} \to \alpha) = 1.$$

Por ejemplo, la regla VP → V NP PP podría tener probabilidad 0.75 en una gramática independiente del contexto dada, y la regla VP → V NP probabilidad 0.25, queriendo decir que es tres veces más probable que una frase verbal (VP) se divida en V NP PP que en V NP. Nótese que las probabilidades de todas las reglas que comparten la misma parte izquierda deben sumar 1. De esta manera, una gramática independiente del contexto es un modelo de análisis sintáctico *generativo*, un modelo de la probabilidad conjunta $P(x, y)$ de la entrada y la salida del proceso de análisis.

En la práctica, podemos extraer una gramática independiente del contexto probabilística a partir de un *treebank* [3], es decir, una colección de oraciones

que han sido anotadas con su correspondiente árbol sintáctico (presumiblemente correcto) por parte de humanos, como puede ser el English Penn Treebank [37]. Esto resulta menos costoso que escribir una gramática a mano, y además nos proporciona una manera sencilla de obtener estimaciones de máxima verosimilitud para las probabilidades de las reglas, a partir de sus frecuencias relativas (por ejemplo, la probabilidad de VP $\rightarrow$ V NP PP se estimará como el número de veces en el treebank que VP se divide en V NP PP, dividido por el número total de apariciones de constituyentes de tipo VP). Una vez que tenemos una gramática independiente del contexto probabilística, se pueden adaptar fácilmente los algoritmos de programación dinámica como CYK o Earley para proporcionar el árbol de análisis más probable para una oración dada. Para ello, se computa la probabilidad de cada árbol como el producto de las probabilidades de las reglas utilizadas, escogiendo el árbol más probable, y resolviendo de esta manera la ambigüedad. Con la ayuda de las mejoras en el hardware conseguidas en los años noventa, esta aproximación produjo los que se pueden considerar primeros analizadores prácticos del lenguaje natural, que producían buenos resultados de precisión sobre oraciones reales [4].

Sin embargo, una debilidad importante de estos modelos son las fuertes suposiciones de independencia que hacen: el modelo generativo supone que la probabilidad de cada posible descomposición de una frase verbal (VP) es independiente del contexto en el cual ésta aparece, lo cual evidentemente no es cierto en el lenguaje real. Para mitigar el problema se han propuesto diversas técnicas de refinamiento de gramáticas, incluyendo la lexicalización [6], la markovización [30] o la división y fusión de categorías [45], consiguiendo mejoras de precisión. Otra opción es utilizar gramáticas suavemente sensibles al contexto, que son formalismos gramaticales más sofisticados que pueden tener en cuenta (hasta cierto punto) el contexto [27], aunque esto tiene un coste en cuanto a complejidad computacional.

Aunque estos analizadores estadísticos dirigidos por gramáticas proporcionaron los primeros resultados prácticos en análisis sintáctico del lenguaje natural, los algoritmos de programación dinámica que utilizan son bastante lentos (con complejidad cúbica en el mejor de los casos, y consiguiendo velocidades de unas pocas oraciones por segundo en la práctica, dado el gran tamaño de las gramáticas extraídas automáticamente que utilizan). A principios de la primera década del siglo XXI, se comenzó a trabajar en aproximaciones más ligeras al análisis sintáctico, dirigiendo cada vez más interés a las representaciones de dependencias (más simples, al no requerir la creación de nodos intermedios durante el análisis) y a modelos puramente dirigidos por los datos que pudiesen ser entrenados directamente sobre los treebanks, sin necesitar una gramática explícita.

En particular, la mayoría de los modelos de este tipo se pueden dividir en dos grandes familias de sistemas que han conseguido buenos resultados y son

ampliamente usadas. La primera de estas familias es la de los *analizadores de dependencias basados en transiciones* [41]. En estos algoritmos, el proceso de análisis se modela como una máquina de estados no determinista. En cada estado, el analizador debe decidir entre diferentes transiciones, que pueden crear dependencias entre palabras, de forma que el proceso completo producirá uno u otro árbol de análisis dependiendo de la secuencia de transiciones que se haya seguido. Para entrenar modelos que puedan escoger el análisis adecuado para cada oración, necesitamos un mecanismo para puntuar las secuencias de transiciones y un algoritmo de búsqueda para intentar encontrar una secuencia de alta puntuación.

En los primeros analizadores basados en transiciones, los modelos de puntuación más populares eran clasificadores como las máquinas de vectores soporte [49] o el perceptrón estructurado [52], entrenados directamente a partir de los análisis de dependencias contenidos en un treebank. El modelo se entrena para dar alta puntuación a aquéllas transiciones que conduzcan a un árbol correcto, obteniendo un sistema final que aproxima un "oráculo" que elige la mejor transición en los árboles del conjunto de entrenamiento. Las características que se pasan como entrada al clasificador pueden capturar información contextual, relajando las suposiciones de independencia con respecto a otros tipos de modelos. Como algoritmo de búsqueda para encontrar una secuencia de transiciones adecuada, se puede usar la búsqueda voraz [41] o la búsqueda en haz (*beam search*) [52] si la velocidad es una prioridad. La programación dinámica [32, 22] también es una opción si se quiere garantizar inferencia exacta (es decir, obtener una secuencia de transiciones con puntuación máxima).

La otra principal aproximación al análisis sintáctico de dependencias dirigido por los datos se llama *análisis basado en grafos*. Los analizadores basados en grafos funcionan puntuando posibles fragmentos del árbol de dependencias y después juntándolos. Esto se puede hacer bien mediante programación dinámica de manera similar a los analizadores basados en gramáticas independientes del contexto probabilísticas, pero sin usar una gramática [10, 46], o bien mediante un algoritmo para el cálculo del árbol de expansión máximo [39].

La flexibilidad de los analizadores de dependencias dirigidos por los datos, su buen equilibrio entre velocidad y precisión, y la reciente aparición de treebanks de dependencias en un gran número de idiomas [42] han convertido a estos modelos en los predominantes en el procesamiento del lenguaje natural. De hecho, muchos de los avances desarrollados primero en analizadores de dependencias dirigidos por los datos se han adaptado después a analizadores de constituyentes: por ejemplo, en la actualidad los modelos basados en transiciones también se utilizan para el análisis de constituyentes, dado que son mucho más rápidos que los modelos basados en gramáticas [35, 12].

Por último, un desarrollo reciente que ha tenido gran impacto tanto en el

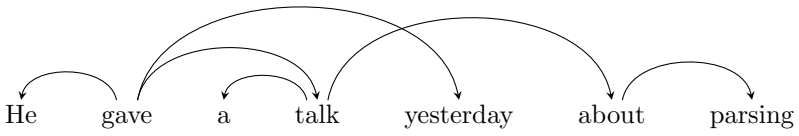

Figure 5: árbol de dependencias para una oración en inglés con dependencias cruzadas.

análisis sintáctico de constituyentes como en el de dependencias es la aplicación de las redes neuronales profundas [36]. Para aplicar estos modelos continuos a las unidades discretas de significado que aparecen en los lenguajes humanos, las palabras se transforman primero en vectores de números reales, llamados *word embeddings* [40]. Esto hace posible usar modelos de aprendizaje profundo para todo tipo de tareas de procesamiento del lenguaje natural, incluyendo el análisis sintáctico. Aunque usar redes "feed-forward" como clasificador para un modelo basado en transiciones proporciona analizadores muy rápidos [5], las mejores cifras de precisión en la actualidad se consiguen con redes neuronales recurrentes, como las LSTMs [23], que proporcionan representaciones vectoriales de las palabras enriquecidas por su contexto en la oración. Dichas redes se han aplicado con éxito para mejorar la precisión de modelos de análisis diversos, incluyendo analizadores de dependencias basados en transiciones [29], analizadores de dependencias basados en árbol de expansión máximo [8], analizadores de dependencias de programación dinámica [22] y analizadores de constituyentes basados en transiciones [34].

3. Desafíos

Gracias a los avances resumidos en la sección anterior, la precisión de los modelos de análisis sintáctico ha venido aumentando hasta un punto en el que es comparable al grado de consenso entre anotadores expertos sobre textos periodísticos en inglés [1]. Sin embargo, los buenos resultados en este caso relativamente fácil no se generalizan a idiomas y dominios más complicados, así que el análisis sintáctico del lenguaje natural está lejos de ser un problema resuelto. Los recursos computacionales que necesitan los algoritmos de análisis, y la búsqueda de modelos psicolingüísticamente plausibles, son otros desafíos a los que se enfrenta este campo de investigación, que se resumen a continuación.

3.1. Analizando idiomas y dominios difíciles

El inglés, el idioma al que tradicionalmente se ha venido dedicando un mayor esfuerzo en la literatura de análisis sintáctico, es un lenguaje atípicamente sencillo de analizar. Su orden de palabras relativamente inflexible y su morfología

simple, junto con la abundancia de recursos y datos de entrenamiento, hacen que sea relativamente fácil conseguir alta precisión en modelos de análisis para el inglés, especialmente en dominios donde los textos tienden a estar escritos en lenguaje estándar que sigue las normas gramaticales, como es el dominio periodístico. Sin embargo, existen varios factores que hacen que algunos idiomas sean mucho más difíciles de analizar:

- Dependencias cruzadas: el inglés tiene una proporción relativamente baja de dependencias cruzadas, aquéllas cuyas flechas se cruzan cuando se dibujan sobre las palabras de la oración, como sucede con la dependencia entre "gave" y "yesterday" y la dependencia entre "talk" y "about" en la Figura 5. Por este motivo, muchos analizadores diseñados para el inglés (o para otros idiomas con esta misma característica, como pueden ser el chino o el japonés) son *proyectivos*, es decir, no pueden construir dependencias cruzadas en absoluto. Esto hace que estos algoritmos sean más eficientes, dado que soportar dependencias cruzadas requiere añadir transiciones o estructuras de datos adicionales a los modelos basados en transiciones [21, 13], aumentar la complejidad computacional de los algoritmos de programación dinámica [19, 22], utilizar gramáticas más complejas [27] o recurrir a los analizadores basados en árbol de expansión máximo [8]. Además de ser más lentos, los algoritmos que soportan dependencias cruzadas tienden a obtener precisiones más bajas debido a la dificultad adicional de aprender este tipo de relaciones.

- Necesidad de segmentación: la división de las oraciones en palabras es una tarea relativamente fácil en inglés, igual que en otros idiomas que utilizan espacios para separar las palabras, ya que éstos resultan de gran ayuda. Sin embargo, idiomas como el chino se escriben sin espacios entre las palabras, haciendo falta aplicar un paso previo de segmentación (división en palabras) que puede introducir ruido adicional en el proceso de análisis sintáctico.

- Morfología rica: ciertos idiomas, como por ejemplo el árabe [11] o el turco [43], son lenguajes sintéticos, es decir, tienen un elevado número de morfemas que modifican el sentido de cada palabra. La morfología tiene una fuerte interacción con la sintaxis, y esta morfología compleja hace que el análisis sintáctico sea notablemente más difícil.

- Textos ruidosos: el uso del lenguaje típico de las redes sociales, como el que se ve en los mensajes de Twitter (tuits), contiene numerosos fenómenos lingüísticos que divergen de la norma gramatical estándar, como pueden ser los emoticonos, los "hashtags" o los frecuentes errores ortográficos, requiriendo técnicas específicas para tratarlos [31].

- Lenguajes con pocos recursos: un factor esencial para obtener una alta precisión en el análisis sintáctico y otras tareas de procesamiento del lenguaje natural es la calidad y la cantidad de los datos de entrenamiento. Algunos idiomas, como pueden ser el inglés, español, alemán o chino, cuentan con cantidades relativamente grandes de datos anotados que se pueden utilizar para entrenar modelos de análisis sintáctico, pero éste no es el caso de muchos otros idiomas, sobre todo los que tienen un pequeño número de hablantes.

El lector interesado puede consultar [51] para ver las precisiones obtenidas en diferentes idiomas y conjuntos de datos por analizadores del estado del arte actual, participantes en una competición reciente. La precisión varía entre más del 90% de dependencias correctas en varios idiomas, y menos del 30% para el kazajo, una lengua túrquica de morfología sintética para el que los datos de entrenamiento disponibles son muy escasos.

3.2. Analizando más rápido

Aunque los actuales algoritmos de análisis sintáctico son lo suficientemente precisos para ser útiles en aplicaciones finales, al menos para algunos idiomas y dominios, sus requerimientos en cuanto a tiempo de computación son todavía un importante obstáculo que limita la adopción generalizada de esta tecnología y la extensión de sus aplicaciones. Los analizadores más precisos basados en gramáticas de constituyentes procesan menos de 5 oraciones por segundo en CPUs recientes [33], mientras que los analizadores neuronales dirigidos por los datos consiguen velocidades de unas pocas decenas de oraciones por segundo (véanse, por ejemplo, las cifras de [12]). El análisis de dependencias puede ser algo más rápido, pero de todos modos, es difícil conseguir más de 100 oraciones por segundo en los modelos neuronales recientes sin recurrir a hardware paralelo [29].

Estas velocidades pueden ser suficientes para aplicaciones que requieren solamente analizar una o unas pocas oraciones de cada vez, como los sistemas de diálogo, pero resultan prohibitivas para el análisis sintáctico a gran escala (por ejemplo, si se quieren analizar grandes colecciones de documentos obtenidas de Internet). El problema es incluso más serio en idiomas que plantean alguno de los desafíos adicionales explicados más arriba, como por ejemplo una proporción significativa de dependencias cruzadas o una morfología rica, lo cual hace que los requisitos computacionales sean todavía más altos [19].

El proyecto FASTPARSE, financiado por el ERC y actualmente en progreso [20], pretende conseguir analizadores más rápidos, combinando técnicas de informática y matemáticas (utilizando razonamiento basado en casos para reutilizar resultados parciales previos en lugar de volver a analizar subestructuras ya conocidas), ciencia cognitiva (creando modelos de análisis inspirados

en cómo los humanos resolvemos la tarea) y lingüística (analizando patrones en las anotaciones y explotándolos para crear algoritmos más rápidos). De este modo, pretendemos acabar con un cuello de botella fundamental para abrir las aplicaciones de procesamiento del lenguaje natural a su aplicación a gran escala, incluso sin la necesidad de las enormes cantidades de recursos computacionales que solamente las grandes empresas tecnológicas pueden desplegar.

3.3. Plausibilidad psicolingüística

Un tercer desafío relacionado con el análisis sintáctico del lenguaje natural es el de conseguir analizadores que sean psicolingüísticamente plausibles, es decir, que analicen las oraciones de forma similar a como lo hacemos los humanos.

Por un lado, esto es importante porque puede servir para avanzar nuestro conocimiento y comprensión del procesamiento del lenguaje por parte de los humanos, y de la propia evolución del lenguaje. Recientemente, la creciente disponibilidad de datos anotados sintácticamente (treebanks) ha abierto un campo de investigación cuantitativo sobre las propiedades universales de la sintaxis de las lenguas humanas: el análisis de los treebanks ha proporcionado evidencias de que los idiomas tienden a ordenar las palabras de manera que se minimiza la longitud de las dependencias [16], y mostrado la relación entre dicha longitud y la presencia de dependencias cruzadas [14], cuya escasez, dada por supuesto por muchos lingüistas durante décadas, sólo se ha confirmado recientemente con evidencia estadística sólida [15]. Un conocimiento más detallado de los aspectos cuantitativos y estadísticos de la sintaxis humana, que por ahora se halla en su infancia, podría a su vez ser de ayuda para diseñar modelos de análisis sintáctico que se ajustaran mejor a las clases de estructuras que aparecen en las oraciones reales.

Por otra parte, como he observado recientemente en [18], las estrategias de análisis existentes que (incluso involuntariamente) se parecen a los modelos de procesamiento humano del lenguaje, en aspectos como analizar las oraciones de izqueirda a derecha o usar una cantidad restringida de memoria, tienden a proporcionar modelos precisos y eficientes. Parece tener sentido plantearse como hipótesis que, dado que los idiomas humanos evolucionaron junto con la mente humana y de tal manera que los seres humanos pudiesen procesarlos de manera eficiente, debería ser posible obtener excelentes modelos de análisis imitando de forma más cercana el procesamiento humano.

4. Conclusiones

En este artículo, he resumido brevemente el campo de investigación del análisis sintáctico del lenguaje natural, presentando su relevancia práctica, las principales aproximaciones estadísticas y de aprendizaje automático que se han aplicado, y mi visión particular sobre los principales desafíos que ofrece para

su investigación presente y futura, y a los que creo que se deberían dedicar especiales esfuerzos en los próximos años. Estos últimos pueden ser un fértil campo de investigación tanto para informáticos como para lingüistas, científicos cognitivos, matemáticos y estadísticos.

References

[1] Berzak, Y., Huang, Y., Barbu, A., Korhonen, A. y Katz, B. (2016) Anchoring and agreement in syntactic annotations, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, Texas, pp. 2215–2224.

[2] Branavan, S. R. K., Silver, D. y Barzilay, R. (2012). Learning to win by reading manuals in a monte-carlo framework, *J. Artif. Int. Res.* **43**(1): 661–704.

[3] Charniak, E. (1996) . Tree-bank grammars, *Proceedings of the National Conference on Artificial Intelligence*, pp. 1031–1036.

[4] Charniak, E. (2000) . A maximum-entropy-inspired parser, *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 132–139.

[5] Chen, D. y Manning, C. (2014) . A fast and accurate dependency parser using neural networks, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, pp. 740–750.

[6] Collins, M. (2003). Head-driven statistical models for natural language parsing, *Comput. Linguist.* **29**(4): 589–637.

[7] Dahl, Ö. (1980). Some arguments for higher nodes in syntax: a reply to Hudson's 'Constituency and dependency', *Linguistics* **18**: 485–488.

[8] Dozat, T., Qi, P. y Manning, C. D. (2017). Stanford's graph-based neural dependency parser at the conll 2017 shared task, *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Association for Computational Linguistics, Vancouver, Canada, pp. 20–30.

[9] Earley, J. (1970). An efficient context-free parsing algorithm, *Communications of the ACM* **13**(2): 94–102.

[10] Eisner, J. M. (1996). Three new probabilistic models for dependency parsing: An exploration, *Proceedings of the 16th conference on Computational*

linguistics-Volume 1, Association for Computational Linguistics, pp. 340–345.

[11] Farghaly, A. y Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions, *ACM Transactions on Asian Language Information Processing (TALIP)* **8**(4): 14:1–14:22.

[12] Fernández-González, D. y Gómez-Rodríguez, C. (2018a). Faster shift-reduce constituent parsing with a non-binary, bottom-up strategy, *arXiv* **1804.07961 [cs.CL]**.

[13] Fernández-González, D. y Gómez-Rodríguez, C. (2018b) . Non-projective dependency parsing with non-local transitions, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, pp. 693–700.

[14] Ferrer-i-Cancho, R. y Gómez-Rodríguez, C. (2016) . Crossings as a side effect of dependency lengths, *Complexity* **21**(S2): 320–328.

[15] Ferrer-i-Cancho, R., Gómez-Rodríguez, C. y Esteban, J. L. (2018) . Are crossing dependencies really scarce?, *Physica A: Statistical Mechanics and its Applications* **493**: 311–329.

[16] Futrell, R., Mahowald, K. y Gibson, E. (2015) . Large-scale evidence of dependency length minimization in 37 languages, *Proceedings of the National Academy of Sciences* **112**(33): 10336–10341.

[17] Gómez-Rodríguez, C. (2010). *Parsing Schemata for Practical Text Analysis*, Vol. 1 of *Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory*, Imperial College Press.

[18] Gómez-Rodríguez, C. (2016a) . Natural language processing and the Now-or-Never bottleneck, *Behavioral and Brain Sciences* **39**: e74.

[19] Gómez-Rodríguez, C. (2016b) . Restricted non-projectivity: Coverage vs. efficiency, *Comput. Linguist.* **42**(4): 809–817.

[20] Gómez-Rodríguez, C. (2017) . Towards fast natural language parsing: FASTPARSE ERC Starting Grant, *Procesamiento del Lenguaje Natural* **59**: 121–124.

[21] Gómez-Rodríguez, C. y Nivre, J. (2013) . Divisible transition systems and multiplanar dependency parsing, *Comput. Linguist.* **39**(4): 799–845.

[22] Gómez-Rodríguez, C., Shi, T. y Lee, L. (2018). Global transition-based non-projective dependency parsing, *Proceedings of ACL*, Association for Computational Linguistics, Melbourne, Australia, p. (To appear).

[23] Hochreiter, S. y Schmidhuber, J. (1997) . Long short-term memory, *Neural Comput.* **9**(8): 1735–1780.

[24] Hudson, R. A. (2007) . *Language Networks: The New Word Grammar*, Oxford University Press, Oxford, UK.

[25] Joshi, M. y Penstein-Rosé, C. (2009). Generalizing dependency features for opinion mining, *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 313–316.

[26] Kahane, S. y Mazziotta, N. (2015). Syntactic polygraphs. a formalism extending both constituency and dependency, *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, Association for Computational Linguistics, Chicago, USA, pp. 152–164.

[27] Kallmeyer, L. (2010) . *Parsing Beyond Context-Free Grammars*, 1st edn, Springer Publishing Company, Incorporated.

[28] Kasami, T. (1965) . An efficient recognition and syntax algorithm for context-free languages, *Scientific Report AFCRL-65-758*, Air Force Cambridge Research Lab., Bedford, Massachussetts.

[29] Kiperwasser, E. y Goldberg, Y. (2016) . Simple and accurate dependency parsing using bidirectional lstm feature representations, *Transactions of the Association for Computational Linguistics* **4**: 313–327.

[30] Klein, D. y Manning, C. D. (2002). Fast exact inference with a factored model for natural language parsing, *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, MIT Press, Cambridge, MA, USA, pp. 3–10.

[31] Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C. y Smith, N. A. (2014). A dependency parser for tweets, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp. 1001–1012.

[32] Kuhlmann, M., Gómez-Rodríguez, C. y Satta, G. (2011). Dynamic programming algorithms for transition-based dependency parsers, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011)*, Association for Computational Linguistics, Portland, Oregon, USA, pp. 673–682.

[33] Kummerfeld, J. K., Hall, D., Curran, J. R. y Klein, D. (2012). Parser showdown at the wall street corral: An empirical investigation of error types in parser output, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, Jeju Island, Korea, pp. 1048–1059.

[34] Kuncoro, A., Ballesteros, M., Kong, L., Dyer, C., Neubig, G. y Smith, N. A. (2017). What do recurrent neural network grammars learn about syntax?, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Association for Computational Linguistics, Valencia, Spain, pp. 1249–1258.

[35] Liu, J. y Zhang, Y. (2017). In-order transition-based constituent parsing, *arXiv preprint arXiv:1707.05000* .

[36] Manning, C. D. (2015). Computational linguistics and deep learning, *Computational Linguistics* **41**(4): 701–707.

[37] Marcus, M. P., Marcinkiewicz, M. A. y Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank, *Comput. Linguist.* **19**(2): 313–330.

[38] McCord, M. C., Murdock, J. W. y Boguraev, B. (2012). Deep parsing in Watson, *IBM Journal of Research and Development* **56**(3/4): 3:1–3:15.

[39] McDonald, R., Pereira, F., Ribarov, K. y Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms, *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP 2005)*, Association for Computational Linguistics, pp. 523–530.

[40] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. y Dean, J. (2013). Distributed representations of words and phrases and their compositionality, *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, Curran Associates Inc., USA, pp. 3111–3119.

[41] Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing, *Computational Linguistics* **34**(4): 513–553.

[42] Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N. et al. (2016) . Universal dependencies v1: A multilingual treebank collection., *LREC*.

[43] Oflazer, K. (2014) . Turkish and its challenges for language processing, *Lang. Resour. Eval.* **48**(4): 639–653.

[44] Padó, S., Noh, T.-G., Stern, A., Wang, R. y Zanoli, R. (2015). Design and realization of a modular architecture for textual entailment., *Natural Language Engineering* **21**(2): 167–200.

[45] Petrov, S., Barrett, L., Thibaux, R. y Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 433–440.

[46] Pitler, E. (2014). A crossing-sensitive third-order factorization for dependency parsing, *Transactions of the Association for Computational Linguistics* **2**: 41–54.

[47] Song, M., Kim, W. C., Lee, D., Heo, G. E. y Kang, K. Y. (2015). PKDE4J: entity and relation extraction for public knowledge discovery, *Journal of Biomedical Informatics* **57**: 320–332.

[48] Vilares, D., Gómez-Rodríguez, C. y Alonso, M. A. (2017). Universal, unsupervised (rule-based), uncovered sentiment analysis, *Knowledge-Based Systems* **118**: 45 – 55.

[49] Yamada, H. y Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines, *Proceedings of IWPT*, Vol. 3, Nancy, France, pp. 195–206.

[50] Yu, M., Gormley, M. R. y Dredze, M. (2015). Combining word embeddings and feature embeddings for fine-grained relation extraction, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Denver, Colorado, pp. 1374–1379.

[51] Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J. et al. (2017). Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies, *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Association for Computational Linguistics, Vancouver, Canada, pp. 1–19.

[52] Zhang, Y. y Nivre, J. (2011). Transition-based dependency parsing with rich non-local features, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, Association for Computational Linguistics, pp. 188–193.

Acerca del autor

Carlos Gómez-Rodríguez es Profesor Contratado Doctor en la Universidade da Coruña. Su investigación se encuadra en el campo de la lingüística computacional y el procesamiento del lenguaje natural, centrándose sobre todo en análisis sintáctico y sus aplicaciones, y abarcando también otros temas como la minería de opiniones o la evolución de la sintaxis de las lenguas humanas. Es autor de un libro y 90 publicaciones con revisión por pares, incluyendo numerosos artículos en los principales congresos y revistas de lingüística computacional. Es investigador principal de un proyecto estatal y del proyecto europeo FASTPARSE (Fast Natural Language Parsing for Large-Scale NLP), financiado por una Starting Grant del European Research Council.