CONCEPCIÓN Y ELABORACIÓN DE UN SISTEMA DE ETIQUETADO SEMIAUTOMÁTICO PARA *UNDER-RESOURCED LANGUAGES*

JOSÉ LUIS PEMBERTY TAMAYO

Trabajo de grado para obtener el título de:
FILÓLOGO HISPANISTA

Asesor
JORGE MAURICIO MOLINA MEJÍA
Doctor en Informática y Ciencias del Lenguaje

PREGRADO
LETRAS: FILOLOGÍA HISPÁNICA
FACULTAD DE COMUNICACIONES
UNIVERSIDAD DE ANTIOQUIA
2020

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Agradecimientos

A mis padres. A la Universidad de Antioquia y a quienes han sido mis profesores y compañeros en el pregrado. Al semillero Corpus Ex Machina y a sus integrantes por el trabajo y conocimiento que hemos compartido. Muy especialmente, además, a los profesores Jorge Mauricio Molina, Víctor Vallejo y Adriana María Ortiz y a los compañeros Andrés Grajales y María Isabel Marín por la dirección y el apoyo en todo este proceso.

Resumen

Título: Concepción y elaboración de un sistema de etiquetado semiautomático para *under-resourced languages*.

Este trabajo parte del hecho de que la lingüística de corpus y la lingüística computacional están cobrando importancia en la comprensión y el estudio de las lenguas, y que la gran diversidad lingüística de nuestro país y del mundo hace que sea difícil la tarea de crear recursos para su procesamiento automático en muchos casos. En este sentido, se propone y se describe un algoritmo que busca asistir el etiquetado manual a nivel de POS de corpus textuales en lenguas para las que aún no existe el etiquetado automático. Así mismo, se presenta un ejemplo de aplicación de este algoritmo a través de un sencillo programa que puede ser probado por el lector.

Palabras clave: *Under-resourced languages*, POS, etiquetado manual, lingüística de corpus, lingüística computacional.

Abstract

Title*:* Conception and development of a semi-automatic tagging system for under-resourced languages.

This work is based in the fact that corpus linguistics and computational linguistics are gaining importance in the understanding and study of languages and that the great linguistic diversity of our country and the world makes it difficult to create resources for their automatic processing in many cases. In this sense, an algorithm that seeks to assist manual tagging at the POS level of textual corpora in languages for which automatic tagging does not yet exist is proposed and described. Also, an example of application of this algorithm is presented through a simple program that can be tested by the reader.

Keywords: Under-resourced languages, POS, manual tagging, corpus linguistics, computational linguistics.

Contenido

Índice de figuras

Índice de tablas

1. Introducción

En los últimos años ha habido un creciente interés por la lingüística de corpus como método de investigación en todos los niveles de la lengua (Brezina, 2018; Mitkov, 2004; Parodi, 2010), puesto que permite trabajar con muestras cada vez mayores y, por consiguiente, tener una mayor confianza en la representatividad de los resultados obtenidos en los estudios. A su vez, el procesamiento de las grandes cantidades de datos, así como su almacenamiento y recolección, dependen cada vez más de técnicas y procedimientos aportados por la lingüística computacional (Mitkov, 2004), esto debido a que los dispositivos informáticos presentan una gran capacidad que permite una mejor realización de tales tareas.

Dentro de la información lingüística que se recolecta para este tipo de trabajos es muy importante el papel del texto escrito, puesto que su procesamiento computacional es relativamente sencillo en comparación con, por ejemplo, la información audiovisual (Baquero, 2010; Parodi, 2010).

En lo que respecta a los trabajos efectuados con corpus de textos escritos, los procedimientos y procesos que se aplican sobre ellos, para obtener unos resultados, varían dependiendo de las necesidades de cada proyecto de investigación (Parodi, 2010); sin embargo, es bastante común encontrarse con el hecho de que es necesario enriquecer la información textual con las nociones lingüísticas o metalingüísticas que se desea estudiar, utilizando lenguajes especializados que permitan introducir en la computadora los datos necesarios y asociarlos a las características del texto, a este proceso se le conoce con el nombre de etiquetado o *tagging* (McEnery & Hardie, 2011; Parodi, 2010).

Con el fin de mejorar estos procedimientos de etiquetado, se han desarrollado, hasta la fecha, varias herramientas que, de manera confiable, automatizan el reconocimiento y la asignación de nociones lingüísticas a textos escritos, identificando en ellos información relativa a la composición morfosintáctica de las palabras y las oraciones, la correferencia textual o incluso la información semántico-pragmática implicada en el llamado análisis de sentimientos; ejemplos de ello serán referidos en el apartado 3.

De las formas comunes de procesamiento existentes, el POS (*Part-of-Speech*) es la más simple y la más necesaria para pasar a niveles más amplios de la lengua o análisis más profundos (Parodi, 2010); de hecho, varias lenguas como el inglés, el español y el francés, cuentan en la actualidad con herramientas que se encargan de automatizar su realización.

La dificultad que suele encontrarse en este campo de trabajo tiene que ver con el hecho de que no es posible automatizar completamente la tarea del etiquetado de POS para todas las lenguas a un mismo tiempo, puesto que la enorme diferencia que existe entre un sistema lingüístico y otro hace que en ocasiones sea inútil aplicar en uno de dichos sistemas los algoritmos y procedimientos que se han desarrollado para cualquier otro; por lo que para determinados casos se hará necesario desarrollar un software específico, contando con las particularidades de cada lengua.

Teniendo en cuenta que solo en el territorio colombiano, por tomar un ejemplo, conviven más de sesenta lenguas diferentes (González & Rodríguez, 2010), y que la realización de un solo producto de software para el etiquetado de una sola lengua puede tardar muchísimo tiempo y requerir de estudios más profundos en ella, se hace patente la perspectiva de que muchas lenguas en el mundo que ya cuentan con producción escrita no tengan, dentro de mucho tiempo, las herramientas básicas para el análisis automático de sus respectivos corpus textuales.

Las lenguas que no cuentan en la actualidad con los recursos informáticos para la realización de este tipo de procesos son denominadas *under-resourced languages*[1] (a partir de ahora URLa) (Krauwer, 2003) y la principal implicación de esta denominación, en lo que respecta a este trabajo, es el hecho de que los procesos de etiquetado deben realizarse de manera completamente manual; lo que implica el empleo de grandes cantidades de tiempo y recursos humanos. Tal situación pone en desventaja a las lenguas incluidas en este grupo, no solo porque se limita la capacidad de información que puede ser tenida en cuenta en una investigación determinada, sino también debido a lo dispendioso que resulta la adecuación de un corpus, lo que puede disuadir a potenciales estudiosos de realizar sus trabajos a partir de estas (Baquero, 2010).

Ante este panorama, la pregunta que se busca resolver en las siguientes páginas es: ¿A través de cuáles procedimientos y algoritmos puede crearse una herramienta computacional que facilite y automatice al menos una parte del etiquetado manual de textos en las *under-resourced languages*?

[1] En este trabajo se ha elegido utilizar el término en su versión original en lengua inglesa, puesto que no se han encontrado traducciones al español y se considera que no resulta pertinente usar las expresiones más literales como *bajos* o *pocos* recursos, porque estas se pueden prestar para interpretaciones en aspectos ajenos a la definición concreta que aquí se acoge.

Lo que aquí se contestará frente a esa pregunta no busca, por supuesto, resolver completamente todos los problemas de carencia de etiquetado automático de todas las lenguas, sino que pretende brindar un dispositivo informático que pueda ser utilizado por investigadores de diferentes partes del mundo que deseen constituir un corpus textual en alguna de las URLa. Este dispositivo podrá usarse para gestionar y facilitar los procesos de etiquetado manual, buscando la manera de hacer automáticas algunas de sus partes, tales como la recurrencia de ciertos términos, las etiquetas, etc., y la trasposición de las nociones lingüísticas realizadas a lenguajes especializados de etiquetado como es el caso de XML (*Extensible Markup Language*).

En este sentido, la información aportada por el investigador y recolectada con ayuda de la herramienta podría funcionar como un diccionario de etiquetas provisional que podría, además, ser reutilizado en el etiquetado de corpus de la misma lengua o ser usado posteriormente en proyectos más grandes para el desarrollo de etiquetadores completamente automáticos.

En beneficio de este proyecto se encuentra la gran capacidad que aportan las herramientas actuales de programación y la extensión de la computación en todo el mundo, haciendo que sea posible no solamente formular los algoritmos que cumplan con la tarea propuesta, sino también su posible implementación en múltiples entornos y dispositivos. También, como se verá más adelante, varios lenguajes de programación implementan librerías diseñadas específicamente para el procesamiento del lenguaje natural, que ofrecen un sólido punto de partida y aportan elementos básicos para esta tarea. En suma, el proyecto es altamente viable con la tecnología de la que se puede disponer en nuestro contexto, y la documentación sobre las herramientas a utilizar abunda en la red; a esto se complementan los conocimientos adquiridos en los procesos del semillero *Corpus Ex Machina* de la Universidad de Antioquia, en los que se apoya este trabajo.

Ahora bien, las razones por las que es necesario proponer un dispositivo informático que cumpla con las funciones anteriormente señaladas pueden definirse según el contexto local del pregrado en Filología Hispánica o según un contexto macro de estudio de las lenguas:

En primer lugar, en lo que respecta al ámbito local, es necesario tener en cuenta que el pregrado forma a futuros estudiosos de las lenguas y que de él surgen diferentes proyectos

de investigación enfocados en toda la diversidad lingüística colombiana; en esta medida, es necesario que desde nuestra misma universidad surjan las propuestas que resuelvan los problemas a los que se enfrentan esas investigaciones; todo esto reconociendo, además, la importancia de las técnicas y procedimientos de la lingüística de corpus y la lingüística computacional, disciplinas a las que se busca acercar las URLa de una manera eficiente, a partir de la que se puedan desarrollar investigaciones de amplia envergadura en todos los temas concernientes al texto escrito en esas lenguas.

En segundo lugar, se puede hablar de una perspectiva macro, que no solamente piensa en las dificultades de los investigadores locales, sino que se propone para el trabajo con cualquier lengua del mundo que utilice información textual susceptible de ser transcrita en una computadora; en esta medida podemos hablar de un posicionamiento de la Universidad de Antioquia en estos temas por medio del trabajo de sus diferentes grupos y semilleros de investigación. Por otra parte, se puede hablar del posible establecimiento de una red de investigadores dedicados a producir y compartir diccionarios de etiquetas que puedan emplearse para el procesamiento de cada lengua en diferentes proyectos, siendo este un paso hacia el desarrollo de herramientas más potentes en lo que respecta al tema.

Finalmente, se tiene el hecho de que los avances que haya sobre una lengua en su procesamiento computacional no son de interés estrictamente académico, puesto que también en campos de la industria, la comunicación y la didáctica de lenguas se hace muy necesario contar con información lingüística etiquetada y codificada para el desarrollo de procesadores de texto, traductores automáticos, sistemas de chat automáticos y un amplio etcétera (Moreno Sandoval, 1998; Nerbonne, 2007; Tordera Yllescas, 2011). Todas estas aplicaciones hallarían un buen punto de partida en textos etiquetados con el dispositivo que aquí se propone y ayudarían a mantener a las lenguas vivas, en uso y en contacto con la actualidad.

2. Objetivos

2.1. General

Concebir y elaborar un sistema informático que permita el etiquetado semiautomático de *under-resourced languages* a partir del desarrollo de un grupo de algoritmos.

2.2. Específicos

- Identificar las implicaciones de la lingüística de corpus y la lingüística computacional en el estudio de las *under-resourced languages.*
- Establecer un sistema de etiquetas que pueda ser implementado en textos de diferentes lenguas.
- Proponer un formato de archivo que pueda ser leído y escrito por los algoritmos en cualquier computadora para estandarizar la producción y lectura de información asociada a todos los proyectos que con ellos se realicen.

3. Antecedentes

Partiendo de la necesidad que se tiene, en el marco de la lingüística computacional y de corpus, de contar con los elementos necesarios para procesar las lenguas a nivel informático (Krauwer, 2003), un claro antecedente de este trabajo es la existencia del etiquetado automático en ciertas lenguas como el español, el inglés y el francés; que se reconoce como una meta a buscar en el trabajo con las URLa. En este campo se pueden destacar herramientas de software libre utilizadas ampliamente en el medio académico como *TreeTagger* (Schmid, 1994) y *TagAnt* (Anthony, 2015); ambas herramientas se caracterizan por utilizar sistemas internos de reglas que permiten llevar a cabo la tarea de etiquetar automáticamente textos en diferentes lenguas al nivel de *Part of Speech*[2] —POS— (Weisser, 2018).

Un etiquetador, también de libre acceso y capaz de establecer etiquetas en otros niveles de la lengua, como la sintaxis y la semántica, es *Freeling* (Padró, Collado, Reese, Lloberes y Castellón, 2010), que cuenta además con la característica de responder a los lineamientos propuestos por el sistema EAGLES para el etiquetado de diferentes lenguas.

El *Expert Advisory Group on Language Ingineering Standards* (EAGLES) ofrece una serie de normas, sugerencias y códigos conformados por letras y números con el fin de estandarizar los trabajos en etiquetado de lenguas (Leech & Wilson, 1996); en este sentido, el presente trabajo también se acoge a los mismos lineamientos, por lo que son antecedentes importantes, tanto el etiquetador mencionado como el sistema de etiquetas.

Por otra parte, en lo tocante al tema de las URLa es importante mencionar diferentes trabajos que se preocupan por su procesamiento computacional. Pueden citarse mayoritariamente ejemplos en que los casos de aplicación en el campo del etiquetado POS se encuentran relacionados principalmente con el desarrollo de herramientas para lenguas concretas como el vietnamita o el árabe (El-Haj, Kruschwitz & Fox, 2015; Le & Besacier, 2009), o bien con preocupaciones por temas como el reconocimiento del habla (Besacier, Barnard, Karpov & Schultz, 2014) o la construcción de corpus basados en información de internet (Scannell, 2007). En este sentido, aunque los trabajos mencionados aportan información valiosa, su desarrollo no se centra directamente en casos en los que se parte de un corpus de textos que

[2] Aunque en inglés el término hace referencia a la categoría gramatical. Estos etiquetadores permiten la anotación de información adicional. Sobre este tema se profundiza en el apartado 4.3.

es necesario etiquetar manualmente para pasar a procesos más complejos, que son la preocupación concreta de este trabajo.

Teniendo esto en cuenta, se hace necesario mencionar trabajos que no usan el concepto de *under-resourced languages* en su documentación y que tienen también preocupaciones más amplias que asistir el etiquetado manual; pero que constituyen antecedentes más directos con respecto a lo que aquí se trata. Estos trabajos son *FieldWorks Language Explorer* (Moe, 2008) y *Field Linguist's ToolBox* (Buseman & Buseman, 2013); ambos diseñados con la finalidad de gestionar y procesar manualmente corpus en diferentes lenguas, principalmente algunas que coincidirán con las características de las URLa.

Sin embargo, la diferencia entre estas herramientas y la que se presenta en los apartados siguientes es que tienen un enfoque mucho más amplio, que busca no solo el etiquetado de POS sino la construcción de un análisis léxico y gramatical cuyo resultado final es un diccionario de la lengua en cuestión (Rogers, 2010); por esta razón, ambos programas presentan una interfaz bastante compleja que puede obstruir una tarea de etiquetado sencillo y carecen de la estandarización ofrecida por el sistema EAGLES, ya que sus sistemas de etiquetas están más centrados en el estudio lexicográfico y antropológico.

Salvando tales diferencias, las dos herramientas antes señaladas comparten con la que aquí se presenta las siguientes características: a) posibilidad de crear etiquetas propias; b) automatización del etiquetado por medio de la reutilización de etiquetas; c) posibilidad de crear una archivo de salida en XML; y d) intención de que la información pueda ser compartida y utilizada entre diferentes investigadores y proyectos.

4. Marco teórico

4.1. Lingüística computacional y procesamiento del lenguaje natural

La lingüística computacional es una disciplina dentro de la lingüística aplicada (Moreno Sandoval, 1998; Tordera Yllescas, 2011) y su principal interés es "la construcción de sistemas informáticos que procesen realmente estructura lingüística y cuyo objetivo sea la simulación parcial de la capacidad lingüística humana" (Moreno Sandoval, 1998, pp. 29–30). Esta idea de procesamiento y emulación de la capacidad lingüística humana es apoyada por autores como Hausser (2013) y Tordera Yllescas (2011). A su vez, el procesamiento del lenguaje natural se define según su búsqueda de esa misma simulación de la capacidad humana (Sáiz Noeda, 2002) y por ello Tordera Yllescas (2011) plantea que no tendría sentido una distinción entre ambos campos. Apoyando esta visión, autores como Hausser (2013) y Moreno Sandoval (1998) utilizan ambos términos indistintamente en una relación de sinonimia, posición que se acogerá también en adelante para este trabajo.

Partiendo de este punto común en la definición, varios autores tienen, sin embargo, diferentes maneras de delimitar los temas y trabajos que pertenecen a la lingüística computacional; estas posiciones van desde criterios bastante generales como[3]: "Computational Linguistics is an interdisciplinary field concerned with the processing of language by computers"[4] (Mitkov, 2004, p. 15), en la que se incluye cualquier operación informática de procesamiento del lenguaje; hasta posiciones más restrictivas que excluyen trabajos como la digitalización de corpus, el análisis de textos literarios, la traducción automática, la enseñanza de lenguas, etc. (Tordera Yllescas, 2011), dando a la disciplina un trasfondo más teórico que práctico.

En esta dicotomía, el trabajo de Moreno Sandoval (1998) encuentra una posición intermedia, además de que hace explícito el perfil de la disciplina por medio de la enumeración de sus posibles aplicaciones. A grandes rasgos, estas aplicaciones son: a) sistemas que tratan de emular la capacidad humana de procesar lenguas naturales; b) programas de ayuda a la

[3] Son propias las traducciones de citas textuales en lengua extranjera que se presentan a lo largo del trabajo.
[4] "La Lingüística Computacional es un campo interdisciplinario que se preocupa por el procesamiento del lenguaje por medio de las computadoras".

escritura y composición textual y c) enseñanza asistida por ordenador y sistemas que ayudan en las tareas lingüísticas (pp. 27–29). Este último grupo es especialmente importante para este trabajo, puesto que dentro de él se incluyen las herramientas para el manejo y etiquetado de corpus que permiten contar con grandes cantidades de datos lingüísticos estructurados, por lo que en ella se recogen los objetivos que se desarrollarán más adelante.

Se ha insinuado, en los párrafos anteriores, otro elemento común a los diferentes teóricos que han abordado el tema, y se trata de la división de la disciplina en lingüística computacional teórica y aplicada (Hausser, 2013; Mitkov, 2004; Moreno Sandoval, 1998; Nerbonne, 2007; Sáiz Noeda, 2002; Tordera Yllescas, 2011). La primera de ellas tiene por objetivo la construcción de abstracciones lingüísticas que puedan abarcar los fenómenos tanto desde el punto de vista informático como desde el punto de vista del lenguaje natural. Por lo tanto, también se encargaría de la formulación de los algoritmos computacionales que ayuden a modelar y poner a prueba esas abstracciones de una manera adecuada (Nerbonne, 2007, p. 3).

Esta primera división resulta importante no solo para la lingüística computacional y la informática, sino que beneficia también a la lingüística en general. Esto se debe a que la abstracción de los conceptos lingüísticos requerida para el procesamiento computacional es susceptible de brindar avances importantes en la manera como se comprende el comportamiento de la lengua en su expresión natural. Esta idea es apoyada por Hausser (2013, p. XXIII), quien asevera además que:

> investigating the particular properties of natural language communication by humans is meaningful only after the mechanism of natural language communication has been understood in principle, modeled computationally, and proven successful in concrete applications on massive amounts of data[5] (p. XXIII).

Esta perspectiva sitúa a la lingüística computacional en un lugar importante en los estudios del lenguaje natural, además de que plantea un futuro prometedor para la disciplina, puesto

[5] "investigar las propiedades particulares de la comunicación del lenguaje natural en los humanos solo tiene sentido después de que el mecanismo de la comunicación del lenguaje natural haya sido entendido en principio, modelado computacionalmente y probado de manera exitosa en aplicaciones concretas en cantidades masivas de datos"

que, para el autor, el procesamiento del lenguaje natural no solo puede producir conocimiento, sino que también su avance se hace necesario para lograr un mayor entendimiento de la capacidad comunicativa humana.

En lo que respecta a la segunda división, la lingüística computacional aplicada abarcaría en general la creación de diversos programas y algoritmos útiles para la manipulación del lenguaje con diversos fines (Nerbonne, 2007, p. 3). Como se ha visto anteriormente, estos fines dependen, en cada autor, de su propia delimitación de lo que atañe a la disciplina; sin embargo, la propuesta de Moreno Sandoval (1998), en la clasificación referida anteriormente, incluye aplicaciones como: a) Traducción automática; b) recuperación de información; c) interfaces hombre-máquina; d) herramientas de análisis textual; e) bases de datos lexicográficas; f) correctores de ortografía, sintaxis y estilo; y g) programas educativos para la enseñanza de lenguas (pp. 27–29).

Por su parte, Nerbonne (2007) ofrece una enumeración menos organizada de diferentes aplicaciones, incluyendo algunas más actuales con respecto a las mencionadas por Moreno como: Reconocimiento del habla, síntesis de voz, minería de datos, sistemas telefónicos automáticos, gestión de documentos académicos y bases de datos, etc. (p. 3)

Como puede verse en los ejemplos citados, el tránsito entre la lingüística computacional teórica y la aplicada es constante y cada una de las divisiones depende de su comunicación con la otra. Por esto, esas designaciones tienen más un valor teórico que ayuda a comprender la amplitud de la disciplina; pero ambas deben ser entendidas en partes iguales dentro de los procesos de producción y aplicación del conocimiento que se adelantan en el procesamiento del lenguaje natural en un sentido general (Hausser, 2013; Nerbonne, 2007).

4.2. Lingüística de corpus

La lingüística de corpus puede definirse, en términos generales, como una “metodología para la investigación de las lenguas y del lenguaje, la cual permite llevar a cabo investigaciones empíricas en contextos auténticos” (Parodi, 2010, p. 15).

Dentro de esta definición, es necesario hacer un especial énfasis en el carácter empírico de la disciplina y auténtico del objeto de estudio; puesto que el punto de partida para los

trabajos de corpus es el modelo funcionalista, que se caracteriza por comprender los fenómenos lingüísticos en situaciones reales de comunicación. Esta perspectiva se opone a la del modelo generativista, que se dedica más bien a teorizar sobre los fenómenos a través de la intuición lingüística (Baquero, 2010, p. 25; McEnery & Hardie, 2013). De ahí la importancia de la autenticidad, que aboga no solo por la conformación de los corpus a partir de datos reales, sino también íntegros, es decir que no se encuentren fragmentados, inconexos o incompletos (Parodi, 2010, p. 15).

Muchos autores (Baquero, 2010; Bernal e Hincapié, 2018; McEnery & Hardie, 2011; Mitkov, 2004) agregan a esta definición la característica de que a la lingüística de corpus le compete la recolección, el procesamiento y el análisis de grandes cantidades de datos; que sean representativos de la lengua o las lenguas que se pretende estudiar.

Ahora bien, el carácter empírico, auténtico y representativo de los datos puede ser utilizado para estudiar la lengua en cualquiera de sus diferentes niveles, como el fonético, el sintáctico o el semántico; o puede también servir a disciplinas de la lingüística aplicada (Parodi, 2010, p. 15). Esta potencialidad dota a la lingüística de corpus de una marcada interdisciplinariedad y hace que su campo de aplicaciones sea muy amplio.

Teniendo en cuenta todo el campo que puede abarcar la lingüística de corpus y su necesidad de trabajar con grandes cantidades de datos y muchas veces procesarlos estadísticamente, se hace notoria su relación con la lingüística computacional, puesto que esta última aporta las herramientas que se requieren para dar un manejo practicable y oportuno a las muestras con las que se busca conformar el corpus, así como sirven, además, para hacer públicos los resultados y ponerlos al servicio de otros investigadores. Esta estrecha relación entre los medios informáticos y la lingüística de corpus es señalada por diversos autores (Baquero, 2010; Bernal e Hincapié, 2018; Parodi, 2010; Tognini-Bonelli, 2001) e incluso hay casos como el de McEnery & Hardie (2011), quienes afirman que el corpus debe estar conformado por "some set of machine-readable texts"[6] (p. 1), dando por sentada la necesidad de los medios informáticos para la conformación del corpus.

Esta relación, tan importante en nuestros días, no estaba tan clara en los comienzos de la lingüística de corpus. Baquero (2010, p. 26) sitúa el comienzo de la lingüística corpus en 1967; sin embargo, pueden reconocerse antecedentes en la década de los 50 (Bernal e

[6] "un conjunto de textos legibles para la máquina."

Hincapié, 2018, p. 12) e incluso en los trabajos realizados con lenguas clásicas en el siglo XIX (Baquero, 2010).

La importancia de estos antecedentes se encuentra en que la carencia de medios informáticos de procesamiento implicaba notables complicaciones como la demanda de un enorme capital en tiempo y trabajo humano, los riesgos del soporte en papel, las dificultades para mantener ordenados y en uso grandes archivos y el peligro que implica la posibilidad del error humano (Baquero, 2010, p. 27; Bernal e Hincapié, 2018, p. 12).

Los medios actuales han permitido salvar esos escollos en gran medida; sin embargo, sus herramientas han favorecido más a unas lenguas que a otras y, tal como lo señala Baquero, hay lenguas que todavía no pueden contar con corpus de tal naturaleza (2010, p. 28), lo que imposibilita su trabajo en el mismo nivel de complejidad y obliga a los investigadores a trabajar con recursos manuales propios de las épocas anteriores a la computación.

4.2.1. ¿Qué es un corpus?

Es necesario, en este punto, precisar lo que se entiende por corpus, puesto que la concepción que se tenga de este término puede determinar en diversos sentidos la manera en que se entienda la metodología de la lingüística de corpus en general.

En principio, siguiendo a Bernal e Hincapié (2018), un corpus es "un conjunto de textos en formato digital […] recolectados, almacenados y sistematizados de acuerdo con criterios lingüísticos" (p. 14); más adelante, los mismos autores agregan que: "Lo que diferencia principalmente un corpus de otras colecciones de textos son los criterios de selección y sistematización, los cuales se ven reflejados en la información que acompaña los datos lingüísticos. Los criterios pueden ser externos e internos" (p. 14).

Como vemos en esta definición, no solo es importante la relación con los medios computacionales, sino que la consideración de un conjunto de textos como corpus tiene que ver con el hecho de que se tenga un criterio claro de selección y que este sea acorde con la investigación que se busca realizar. Para lograr esta adecuación se encuentran en el trabajo de Parodi (2010) unas características más específicas que deben ser cumplidas por el conjunto de textos; estas son:

a. Recolección de textos en entornos naturales
b. Explicitud de los rasgos definitorios compartidos por los textos constitutivos

c. Formato final de tipo digital plano (*.txt) para cada texto o documento
d. Tamaño, preferentemente, extenso
e. Respeto a principios ecológicos
f. Etiquetaje computacional semi-automático de naturaleza morfosintáctica u otra para cada texto
g. Disponibilidad a través de medios computacionales
h. Acceso a visualización completa de los textos que lo componen en formato plano
i. Búsqueda de principios de proporcionalidad o representatividad (posiblemente estadística)
j. Sustento o procedencia inicial especificada
k. Identificación de una organización en torno a temas, tipos de textos, registros, géneros, etc.
l. Registro de datos cuantitativos que permita la comparación y posible normalización de cifras (p. 26).[7]

Sin embargo, el mismo autor plantea más adelante que no es necesario que un corpus cumpla estrictamente con todas estas características, puesto que las diferentes investigaciones pueden requerir solamente algunas de ellas para lograr adecuadamente sus objetivos (p. 27).

Teniendo todo esto en cuenta, los corpus pueden ser clasificados según diferentes criterios. Aquí se seguirá el planteamiento de McEnery & Hardie (2011), quienes proponen los siguientes:

- Mode of comunication
- Data collection regime
- Total accountability versus data selection
- Multilingual versus monolingual corpora
- Corpus based- versus corpus driven linguistics
- The use of annotated versus unannotated corpora (p. 3).[8]

Según el primer criterio, los elementos que componen el corpus pueden ser escritos, orales o de otros tipos. El segundo se refiere a las diferentes maneras de recoger el corpus; dentro de estas se destacaría el corpus monitor, que crece con el tiempo a medida que pueden encontrarse nuevos elementos para enriquecerlo y el corpus de muestra, que selecciona una muestra estable según los criterios necesarios para la investigación; también los autores tienen en cuenta un tipo de corpus *oportunista*, que se conforma recolectando todos los textos posibles en casos en que una lengua, tema o variedad tengan poca cantidad de ellos o haya mucha dificultad para conseguirlos.

[7] La lista aparece en el original con numerales en lugar de literales.
[8] "Modo de comunicación / Régimen de recolección de datos / Responsabilidad total vs. selección de datos / Corpus multilingüe vs. monolingüe / Lingüística basada en corpus vs. dirigida por el corpus / El uso de corpus anotado vs. no anotado"

El tercer criterio hace referencia a que, en principio, el conjunto de textos debe representar una totalidad y no ser selectivo solo con los ejemplos que apoyan una determinada teoría; sin embargo, también se plantea que la selección puede ser funcional en algunos casos. El cuarto se refiere a la cantidad de lenguas de las que procedan los textos que conforman el conjunto, que pueden ser una o varias, dando como resultado diferentes posibilidades de análisis.

El quinto criterio de clasificación hace referencia a un punto importante en la concepción del corpus y es abordado por otros autores ya citados (Bernal e Hincapié, 2018; Tognini-Bonelli, 2001) y se refiere a la diferencia que hay entre las aproximaciones basada en corpus o dirigida por el corpus (*corpus-based* y *corpus-driven*, por su denominación en inglés) en el estudio de la lengua. La primera de ellas se refiere a una investigación en la que se parte de una hipótesis y se recoge un corpus específicamente para responder a las necesidades de la comprobación de esa hipótesis; es normal que este tipo de corpus sea más pequeño y cuente con el etiquetado de información muy específica. La segunda aproximación hace referencia a la recolección previa de un corpus, en cuya observación y estudio pueden encontrarse fenómenos y patrones que podrían ser difíciles de notar de otra forma y que generan hipótesis e investigaciones a partir de allí; los corpus realizados con este propósito tienden a ser de un tamaño mayor y a estar en constante crecimiento, así como a tener un etiquetado más básico y general de tipo morfológico y sintáctico.

El último criterio propuesto por McEnery & Hardie (2011) hace referencia, precisamente, al etiquetado o anotación de un corpus. Este es un proceso que consiste en introducir un análisis lingüístico sobre los elementos del texto que dé cuenta de sus características en cualquiera de los niveles de la lengua. Este análisis, según los autores, puede introducirse en el texto editándolo o codificándolo de alguna manera o bien puede ser mantenido aparte, pero igualmente ligado a los datos del corpus. Así mismo, algunos investigadores pueden preferir evitar el etiquetado, dando cuenta de su análisis de otras maneras; sin embargo, aunque el corpus no se encuentre ligado a un sistema de codificación del análisis, este siempre estará presente en otros formatos. Sobre este tema del etiquetado de corpus se hablará en el apartado siguiente.

4.3. Etiquetado de corpus

En consonancia con lo mencionado en el último párrafo del apartado anterior y con algunos de los criterios tomados de Parodi (2010), en el trabajo de Hincapié y Bernal (2018) se proponen las siguientes fases para la constitución de un corpus:

a) El diseño de corpus.
b) La obtención de permisos y captura de datos.
c) La planeación y preparación del sistema de almacenamiento.
d) El procesamiento del corpus.
e) Las opciones de uso (p. 53).

De ellas, la más importante para este trabajo es la de *Procesamiento del corpus*, puesto que es en esta fase donde se convierten los caracteres del lenguaje normal a un sistema de codificación que pueda ser procesado por medio de sistemas informáticos (Bernal e Hincapié, 2018, p. 57).

Al mencionado proceso de conversión es a lo que se refieren diferentes autores con distintos términos que se entienden aquí como sinónimos y son: *etiquetado, anotación, etiquetaje, marcaje y tagging*. Una definición formal puede encontrarse en el trabajo de McEnery & Hardie cuando dicen que: "corpus annotation is largely the process of providing – in a systematic and accesible form – those analyses which a linguist would, in all likelihood, carry out anyway on whatever data they worked with"[9] (2011, p. 13).

De esta definición es importante señalar dos aspectos, el primero de ellos es la necesidad de una manera sistemática y accesible de etiquetar, para lo que es común la utilización de lenguajes informáticos como el XML, el HTML y el GML (Bernal e Hincapié, 2018, p. 57). El segundo aspecto es la amplitud de la información que puede ser etiquetada, que depende de las necesidades que el investigador tenga a la hora de construir el corpus.

Sin embargo, dentro de tal amplitud se han podido fijar, hasta el momento, ciertos elementos que son comunes en el etiquetado de corpus y que constituyen un punto de partida para procesos más complejos. En este caso es necesario referirse al trabajo de Parodi, quien

[9] "La anotación de corpus es en gran parte el proceso de proporcionar –de una manera sistemática y accesible– aquellos análisis que un lingüista, con toda probabilidad, llevaría a cabo de todos modos con los datos con los que trabajara"

identifica una diferencia entre el marcaje estructural y la anotación lingüística; siendo el primero la introducción de información que podemos llamar paralingüística, como la procedencia del texto, su autor, su extensión, un código que lo identifique dentro del conjunto, la identificación de hablantes, etc. (Parodi, 2010, p. 39). A su vez, la anotación lingüística se refiere, tanto para el mismo Parodi como para McEnery & Hardie (2011), a toda aquella información de cualquier nivel de análisis de la lengua que pueda enriquecer el texto, como lo son, por ejemplo, la información fonética, semántica, morfológica o pragmática que se colija de los elementos del texto.

Dentro de la anotación lingüística se han caracterizado también algunas formas de etiquetado que son comunes en el trabajo con corpus, como son el llamado *Part of Speech* y el análisis de tipo *Parser*; este última se centra en un análisis de las funciones que cumple cada palabra en la sintaxis de la oración (Parodi, 2010, p. 40).

Sin embargo, y teniendo en cuenta los objetivos del presente trabajo, se le dará mayor relevancia al etiquetado de tipo POS; otros nombres que pueden referirse al POS son, según Mitkov: "morfological, word class, even lexical"[10] (2004, p. 225). Esta diferencia en la forma de denominarlo se debe al tipo de información que suele resultar de este etiquetado. En primera instancia el término *Part of Speech* se refiere en inglés a lo que conocemos en español como categorías gramaticales. No obstante, como también lo menciona Mitkov, los etiquetadores que se consideran de POS suelen entregar mucha más información que esta, tal como contenidos léxicos, morfológicos y semánticos que van desde el número gramatical hasta la clasificación de los nombres como propios o comunes (2004, p. 225).

Por otra parte, también se han identificado diferentes maneras de llevar a cabo la tarea de etiquetado de los textos que componen un corpus. McEnery & Hardie plantean tres aproximaciones al problema, que serían el etiquetado completamente automático, el etiquetado completamente manual y el etiquetado automático revisado de manera manual. Los tres enfoques poseen cada uno sus ventajas, desventajas y márgenes de error (2011, p. 49).

En el etiquetado automático es necesario destacar la actuación de los programas denominados etiquetadores, que, por medio de una serie de reglas gramaticales, diccionarios y algoritmos, permiten analizar automáticamente un texto y dar como resultado, en muy poco

[10] "morfológico, de clases de palabras e incluso léxico"

tiempo, etiquetados del tipo POS o *Praser*. Ejemplos de esto son algunos de los trabajos mencionados en el apartado 3 (*Freeling, TagAnt* o *Treetagger*).

La automatización de estos procesos, sin embargo, no está exenta de errores (McEnery & Hardie, 2011, p. 31) y, por ello, los autores proponen el etiquetado automático con corrección manual en casos en que se requiera de una exactitud completa en el análisis. Además de este problema, también se encuentra el hecho de que no todos los fenómenos lingüísticos son igualmente fáciles de automatizar y que no todas las lenguas cuentan con programas que cumplan estas funciones; en estos casos se plantea la necesidad de un etiquetado completamente manual, en el que se requiere la inversión de recursos temporales y humanos más altos para realizar el análisis y hacerlo constar en las etiquetas.

En oposición a los avances en las técnicas y niveles de etiquetado, también se ha planteado la postura de que es contraproducente etiquetar los textos, sea porque se impone un análisis al corpus o porque las etiquetas asignadas pueden no ser las adecuadas; en ambos casos lo que se defiende es la pureza del texto (Sinclair, 1992). Ante estas objeciones, sin embargo, es pertinente referirse a las palabras de McEnery & Hardie (2011) cuando dicen que la anotación es algo que se hace siempre y desde hace mucho en el análisis de corpus, pero que simplemente no se hace explícita en algunos casos (p. 13) y que no se trata de un proceso que añada o reste nada a la información original, puesto que solamente se dedica a hacer constar información que ya se encuentra implícita en ese uso particular de la lengua, por lo que más bien se habla de enriquecer los datos de los que puede disponer un sistema (p. 31).

4.4. Under-resourced languages

Ya han sido mencionadas las diferentes aplicaciones de la lingüística computacional en diversos ámbitos y la conveniencia de los corpus debidamente anotados para tales objetivos, así como la importancia que pueden tener en la eficacia de estos procesos las herramientas informáticas adecuadas para su automatización.

Teniendo esto en cuenta, la disponibilidad de recursos informáticos para el procesamiento de una lengua (entre otros varios motivos de carácter social, demográfico, económico

y político) es un factor bastante determinante en la elección de un investigador a la hora de tomarla como objeto de estudio (Maxwell & Hughes, 2006, p. 29).

Ante este panorama, las *under-resourced languages* (URLa) pueden definirse de manera amplia como aquellas lenguas que en la actualidad no disponen de los programas informáticos o los componentes de corpus que las hagan accesibles a las aplicaciones de la lingüística computacional, tales como la traducción automática o el reconocimiento de voz; lo que limita a sus hablantes en el aprovechamiento de los beneficios de la sociedad de la información (Krauwer, 2003). Algunos sinónimos que pueden encontrarse en otros trabajos son: "*low-density languages, resource-poor languages, lowdata languages, less-resourced languages*" (Besacier et al., 2014, p. 87).[11]

En una definición más concreta, propuesta en primera instancia en los trabajos de Krauwer (2003) y Berment (2004), se aporta una serie de criterios que sirven para caracterizar la categoría en cuestión y son los siguientes:

a) Carencia de un único sistema de escritura o una ortografía estable.
b) Presencia limitada en la web.
c) Carencia de expertos en lingüística.
d) Carencia de recursos electrónicos para el procesamiento del habla y la lengua.
e) Carencia de corpus monolingües.
f) Carencia de diccionarios bilingües electrónicos.
g) Carencia de corpus oral transcrito.
h) Carencia de diccionarios de pronunciación y vocabularios.

Como puede verse en esta lista, la disponibilidad de recursos de una lengua no depende enteramente de la existencia de piezas de software especializadas en ella, sino que requiere también de los recursos lexicográficos, de corpus y de accesibilidad a material lingüístico que pueda ser objeto de investigación, como insumos necesarios para construir las herramientas capaces de procesar la lengua.

Antes de continuar, es necesario apuntar, siguiendo a Besacier et al. (2014), que el concepto de URLa no es equivalente al de lengua minoritaria, puesto que hay ejemplos como el catalán, que, siendo minoritaria en España, cuenta con una amplia cantidad de textos accesibles y con recursos como el de la traducción automática. Por otra parte, es necesario

[11] Continuando la lógica con respecto a no traducir el término *under-resourced languages*, tampoco serán traducidos estos sinónimos.

reconocer que es común encontrar que una lengua pertenece a ambos grupos al mismo tiempo (p. 87).

Tal y como lo mencionan algunos autores, llevar un control del estado actual de las lenguas en términos de aplicaciones de la lingüística computacional no es fácil, puesto que el panorama se encuentra en constante cambio y muchas veces es imposible conocer las investigaciones en curso (Besacier et al., 2014; Maxwell & Hughes, 2006). Sin embargo, los mismos autores refieren diferentes estándares propuestos para la medición de una escala en la que puedan ubicarse las diferentes lenguas, entre los que destaca el llamado BLARK.

El BLARK o *Basic Language Resource Kit*[12] (Krauwer, 2003, p. 4) es una propuesta de diferentes organismos europeos para formalizar una lista de recursos que deben ser la primera meta para los investigadores en las URLa y mantener, de ese modo, una igualdad en el acceso a los beneficios del procesamiento del lenguaje natural. Sus elementos, tal y como los describe Krauwer, consisten básicamente en subsanar las carencias descritas anteriormente como características particulares de este grupo de lenguas.

Cabe resaltar que tanto el BLARK como otras caracterizaciones mencionadas en los trabajos citados, están basadas en investigaciones de Europa y enfocadas hacia su panorama lingüístico. Por tal motivo, aunque pretenden ser en gran medida abiertas para cualquier lengua del mundo, es posible que dejen de lado las características que puedan hacer particular alguna zona, lengua o grupo de lenguas en el resto del mundo. A causa de esto, además, la búsqueda de literatura al respecto revela un vacío en la actualidad con respecto a este tema en Latinoamérica y, más particularmente, en Colombia.

[12] Equipo básico de recursos de la lengua.

5. Metodología

5.1. Nociones previas

Como se ha evidenciado en los apartados anteriores, los estudios de corpus en la actualidad son de gran interés para diferentes aspectos de cada lengua y se encuentran estrechamente ligados con la capacidad de procesar los datos lingüísticos de manera computacional.

Entendiendo esto, las páginas siguientes pretenden proponer un algoritmo que pueda utilizarse para la creación de un software orientado a asistir los procesos de etiquetado manual de corpus, enfocado al trabajo con lenguas que no cuentan con la posibilidad del etiquetado automático. Se entiende por algoritmo lo siguiente: "a set of steps to accomplish a task that is described precisely enough that a computer can run it"[13] (Cormen, 2013, p. 1); esta serie de pasos estaría orientada a que una entrada de datos determinada realice con estos una serie de procesos que permitan producir una salida de datos, con la información o resultados que un usuario espera recibir (Cormen, 2013, p. 2).

En principio, y como se señaló anteriormente, el nivel de etiquetado que se trabajará es el denominado POS, puesto que es el más común en los trabajos actuales y sienta el punto de partida para procesos mucho más complejos (Maxwell & Hughes, 2006). En este apartado se explicarán algunos elementos generales de la propuesta, para continuar posteriormente con una descripción detallada de los algoritmos que la componen.

A raíz de la elección del POS como enfoque para la realización de este trabajo, se hace necesaria la adopción de un sistema de etiquetas para las diferentes categorías que puedan asignarse a las palabras en este nivel. El sistema con el que se trabajará es el denominado EAGLES (Leech & Wilson, 1996), puesto que gracias a él se busca establecer una estandarización internacional y multilingüe para el etiquetado de este nivel de la lengua, y ya lo tienen en uso varios proyectos como el antes mencionado *Freeling* (Padró et al., 2010). Las etiquetas de EAGLES permiten codificar, a través de una combinación corta de números y

[13] "una serie de pasos para realizar una tarea que están descritos de manera lo suficientemente precisa como para que una computadora pueda ejecutarlos"

letras, datos como la categoría gramatical de la palabra y, dependiendo de esta, el género, número, caso, tiempo, modo o aspecto. A continuación, se muestra un ejemplo:[14]

Tabla 1. *Ejemplo de uso de las etiquetas EAGLES*

YO	COMPRABA	PAN
PP1CSN0	VMII3S0	NCMS000
P = Pronombre	V = Verbo	N = Sustantivo
P = Personal	M = Principal	C = Común
1 = Primera persona	I = Indicativo	M = Masculino
C = De género neutro o indeterminado	I = Pretérito imperfecto	S = Singular
S = Singular	3 = Tercera persona	0 = Podría tomar otro valor si fuera un nombre propio clasificable.
N = Caso nominativo	S = Singular	0 = Valor también asociado a la clasificación de nombres propios
0 = Registro informal	0 = De género neutro o indeterminado	0 = Valor asociado a la presencia de rasgos apreciativos en sustantivos y adjetivos

Como puede observarse en la tabla, el sistema EAGLES cuenta con unas etiquetas que se encuentran conformadas por ciertas letras o números en determinadas posiciones, de manera que la aparición de un carácter en una posición se usa para dar cuenta de un rasgo entre varios posibles, según la categoría gramatical de la palabra. Una descripción más detallada de los valores que puede tomar cada posición se puede observar en el *Anexo A*.

Habiendo dicho esto, se hace necesario especificar una manera en la que las etiquetas asignadas a los componentes de los textos puedan ser vinculadas con estos. Para esta función se utilizará el lenguaje XML, que permite tener un tipo de documento compuesto por diferentes elementos a los que se pueden agregar características; en este caso los elementos serían los componentes del texto y a cada uno se le asignaría la etiqueta pertinente. Para explicar de mejor forma el funcionamiento de este lenguaje se muestra un ejemplo a continuación:

[14] Tanto las figuras como las tablas que en adelante aparecen en el cuerpo de este trabajo son de elaboración propia.

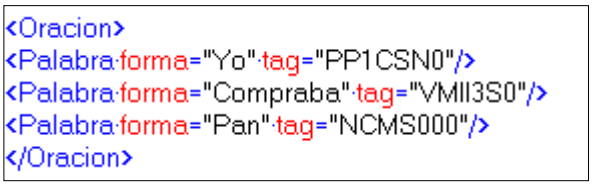

Figura 1. *Ejemplo de uso de lenguaje XML*

En este caso puede verse cómo el lenguaje XML utiliza ciertas convenciones textuales para definir unos elementos y asignar a estos sus características, siendo una de las principales la utilización de los signos < > (conocidos con el nombre de balizas) para marcar el inicio y el final de lo que abarca cada elemento. En color azul pueden verse en la imagen los elementos, que son en este caso uno llamado *Oracion* cuyo contenido son otros tres, cada uno de tipo *Palabra*. En color rojo pueden verse las características de los elementos, conocidas también con el nombre de atributos. Se puede apreciar, además, que el tipo de elemento y el nombre de los atributos no varían para cada palabra, y aquello que cambia son los diferentes valores que pueden tomar esos atributos, esta información se muestra en la imagen en letras negras.

Esta combinación de elementos variables y no variables crea dentro de un documento o un grupo de documentos XML una estandarización que permite a los sistemas informáticos y a los observadores humanos una manera fácil de encontrar la información necesaria cuando hay grandes masas de datos, como puede ser el caso en los textos de un corpus lingüístico.

Utilizando estos elementos básicos y algunos otros más avanzados, los documentos en lenguaje XML pueden tomar formas muy complejas que representan grupos de datos y la manera en que esos datos se relacionan con otros. Debido a esta característica, dicho lenguaje se encuentra en uso en diferentes aplicaciones en el mundo de la informática y tiene un lugar especial en el trabajo de etiquetado y estudio de corpus lingüísticos; siendo común encontrarlo como formato a partir del cual organizan sus datos diferentes etiquetadores automáticos, como los ya mencionados en capítulos anteriores. Un ejemplo de su uso en aplicaciones que se encuentran en la red también puede verse en el etiquetado del corpus PRESEEA Medellín (Molina Mejía et al., 2017) y el corpus DICEELE (Molina Mejía et al., 2019), ambos realizados en el marco de proyectos de investigación en la Universidad de Antioquia.

Teniendo en cuenta estas dos maneras de codificar la información lingüística para hacerla accesible y manejable a través de medios computacionales, la intención del presente trabajo es que luego de utilizar los algoritmos, de los que se hablará en las siguientes páginas, un investigador pueda contar con los textos de un corpus etiquetados en lenguaje XML y con

las etiquetas EAGLES; ambas características harán del corpus un trabajo que se acoge a estándares en vigor y que se presta para procesos posteriores, tanto en la perspectiva de los estudios basados en corpus, como en los dirigidos por el corpus.

5.2. Descripción del programa y sus algoritmos

En este apartado se pasará a describir de manera general el funcionamiento del modelo que se propone; es decir, los procesos que realizará el software una vez programado para asistir a un investigador en el etiquetado manual de los textos que compongan un corpus, así como la automatización de varias tareas que permitirán agilizar esta actividad.

5.2.1. Estructura

La parte principal del programa estará representada por una interfaz constituida por una ventana, que es el medio en que el usuario puede llevar a cabo todas las interacciones con el sistema. A través de esta será posible introducir los textos que conformen el corpus, visualizar cada una de las unidades que serán etiquetadas y asignarles las etiquetas pertinentes. Esta interfaz también estará respaldada por un sistema de archivos y carpetas que se ocuparán de almacenar la información necesaria.

Entre esos archivos y carpetas, el programa leerá y escribirá datos constantemente. En primer lugar, el sistema contará con una carpeta utilizada para almacenar los diccionarios, es decir, toda la información introducida por el usuario sobre las unidades etiquetadas; esto con el fin de poder automatizar el etiquetado de las mismas unidades en momentos posteriores, en el mismo corpus o en otro donde tenga presencia esa unidad. En esta carpeta podrán guardarse múltiples diccionarios, que estarán asociados a diferentes proyectos de etiquetado o a otros que provengan de trabajos previos con la misma lengua que hayan compartido sus avances. En segundo lugar, habrá también una carpeta en la que se guardarán los archivos resultantes del proceso, que podrá ser elegida por el usuario.

Además de los archivos de diccionario y los archivos de salida con los textos etiquetados, el programa llevará cuenta de sus avances en un determinado corpus y un texto específico de ese corpus, con el fin de consignarlos en otro archivo. Toda la información guardada permitirá interrumpir el proceso de etiquetado en cualquier momento sin perder los avances

y poder retomarlo posteriormente, teniendo en cuenta que el etiquetado manual de un corpus, aún con la ayuda que brindará este sistema, puede ser una tarea larga y dispendiosa.

5.2.2. Etiquetado

Para llevar a cabo el proceso de etiquetado de un corpus, un usuario deberá contar con todos los textos organizados en formato de texto plano (*.txt) en una carpeta determinada. Al inicio de la sesión, el programa permitirá que se ingrese la dirección de la carpeta que contiene los textos y guardará sus nombres en una lista ordenada; esto con el fin de seguir el avance del etiquetado a medida que pase por cada uno de los textos.

Posteriormente, el programa leerá el primer texto y guardará su contenido. La primera acción que se realizará con el contenido del texto será separarlo por palabras o *tokens*; este proceso es común en el medio del procesamiento de corpus y se explicará de manera detallada en el apartado 6.2. Cada una de las palabras será almacenada por el programa en una lista que permita ser recorrida de forma ordenada.

Contando con la lista de palabras, el programa procederá a mostrar, en su ventana principal, cada una de ellas, agregando antes y después las palabras que aparecen junto a ella para que no esté descontextualizada. Viendo la palabra con su contexto, el usuario podrá elegir de una lista de opciones que coincidirán con las características de las etiquetas EAGLES; primero seleccionará la categoría gramatical de la palabra y, dependiendo de ella, opciones adicionales como las señaladas en la Tabla 1. Esta elección de opciones se realizará por medio de los nombres de las categorías, como *sustantivo, plural* o *tercera persona*, por lo que el investigador no necesitará conocer con precisión los códigos usados en el sistema EAGLES, y podrá, por lo tanto, confiar su interpretación a los sistemas informáticos.

En este punto también estará disponible un espacio en que el usuario pueda editar la etiqueta generada por el sistema con base en sus elecciones o pueda escribir su propia etiqueta en caso de que desee trabajar con nociones diferentes a las propuestas en primera instancia. Gracias a esta opción, el sistema permitirá etiquetar a partir de nociones diferentes a las morfosintácticas, siendo el investigador quien elige cómo desea fijar sus etiquetas según sus objetivos.

Habiendo fijado una etiqueta para la palabra, el usuario podrá elegir entre dos opciones; la primera de ellas se encarga simplemente de generar una línea de código XML en el archivo que tendrá finalmente las etiquetas de todo el texto. La segunda opción, además de

escribir esa línea en el archivo de salida, generará una entrada en el archivo del diccionario, donde conste la palabra y la etiqueta que se le asignó.

Cada vez que el sistema pase a una nueva palabra, revisará el archivo del diccionario para verificar si hay una entrada que le corresponda; de ser así, pasará a escribir una entrada correspondiente en el archivo XML sin que el usuario tenga que introducir de nuevo la etiqueta. De esta manera, cuando el usuario reconozca una palabra que llevará siempre la misma etiqueta (por ejemplo, una que siempre podrá reconocerse como sustantivo singular masculino) el sistema se encargará de etiquetarla automáticamente.

Partiendo de este procedimiento, a medida que el usuario avance en el etiquetado de las palabras de un texto, irá creando un diccionario más robusto que le permitirá avanzar más rápidamente en el resto del texto. Esto aplica también para el resto de los textos del corpus, pues el mismo diccionario será utilizado para todos ellos y siempre se buscarán en él las palabras antes de consultar con el usuario.

Cuando una palabra pueda tener diferentes etiquetas según el caso, el usuario podrá utilizar la primera opción para no generar una entrada en el diccionario y lograr así que el sistema siempre le solicite introducir la etiqueta adecuada.

5.2.3. Recuperación y reutilización

A medida que se llevan a cabo los procesos descritos anteriormente, el programa llevará la cuenta de su avance, indicando exactamente en qué palabra de qué texto se encuentra en cada momento. Estos datos serán consignados en el archivo que anteriormente se describió para este fin.

Teniendo como referente este archivo, cada vez que el usuario inicie el programa, tendrá la posibilidad de recuperar su información para que el sistema lo sitúe en el lugar exacto en el que interrumpió un proceso anterior, permitiéndole continuar sin que se vea afectado ninguno de los archivos de salida o de diccionario. Este es el proceso que se entiende en este caso como *recuperación*.

Sin embargo, otro tipo de recuperación será permitido por el programa, y este puede tener implicaciones más importantes: al etiquetar un grupo grande de textos, a partir de los procesos ya mencionados, un investigador puede estar generando un diccionario robusto de una lengua determinada que contenga los datos necesarios para agilizar procesos en la misma

lengua para estudios u objetivos diferentes. De esta manera, el sistema permitirá que un archivo de diccionario pueda ser reutilizado, simplemente seleccionándolo en la interfaz del programa, lo que facilitará la creación de redes de cooperación entre investigadores de una misma lengua y permitirá trabajos posteriores para la completa automatización del etiquetado.

5.3. Aportes y limitaciones

En el marco de la definición que se ha dado anteriormente de la lingüística computacional, puede entenderse esta propuesta como un trabajo de lingüística aplicada con un carácter interdisciplinario hacia las ciencias del procesamiento de la información. Tomando esto en consideración, se puede inferir que gran parte de los procesos que aquí se describen están enfocados a la resolución de problemas propios de la lingüística de corpus en el marco de las URLa.

Con esto en mente, el principal aporte del presente trabajo es proponer una serie de pasos que puedan ser utilizados para reducir el tiempo y el trabajo necesarios en la constitución de un corpus etiquetado en el nivel de POS, tratándose de lenguas que no cuenten con los recursos para su procesamiento automático en este campo.

Sin embargo, la manera en que el sistema permite la construcción de etiquetas propias y su reutilización en casos similares se presta para proyectos de otros tipos. Supóngase, por ejemplo, que un investigador desea realizar un proceso de etiquetado manual que evidencie el uso de una colocación determinada, de una expresión compuesta o de una paremia. En todos estos casos el algoritmo propuesto sería útil, pues el investigador podría buscar en el texto las unidades que necesita y crear etiquetas para ellas, guardándolas para que sean usadas cada vez que se encuentre esa misma unidad en el corpus. Para excluir las unidades que no sean de interés para el proyecto, pueden dejarse simplemente con etiquetas en blanco.

Otra manera en la que podrían cumplirse tareas como las antes descritas es, si se conoce de antemano una lista de unidades a buscar, editar manualmente un diccionario con el formato especificado en el apartado 6.3 y utilizarlo para que el programa encuentre y etiquete automáticamente los casos deseados dentro del grupo de textos.

Así, las características principales del modelo descrito hacen que sea posible utilizarlo siempre que se requiera un etiquetado manual a un nivel que esté por encima del de la palabra individual, no solamente en las URLa, sino también en tareas específicas de lenguas que no estarían dentro de este grupo.

Por otra parte, no es posible esperar de este modelo una automatización total del proceso de etiquetado POS a la manera de los etiquetadores mencionados en el apartado de *Antecedentes*; puesto que el sistema de archivos de diccionario depende mucho de las características que sean introducidas de forma manual y siempre será necesario prestar atención a la aparición de casos nuevos o a la resolución de casos que no puedan guardarse en el diccionario por ser ambiguos.

Además, a pesar de que el enfoque de este trabajo pretende ser amplio en términos de las lenguas que pueden ser utilizadas, este grupo se limita a aquellas que se escriban con los caracteres comprendidos en la codificación UTF-8 y que se construyan por medio de la utilización de palabras o *tokens* separados por espacios. Esto no excluye, sin embargo, que el mismo sistema de guardado y reutilización de información pueda ser empleado por otros investigadores para constituir sistemas parecidos que permitan el uso de otros caracteres u otras formas de escritura.

6. Estructura detallada del programa

Como resultado de todas estas consideraciones, y a manera de solución al problema planteado al comienzo del trabajo, se describirá a continuación el *UnderRL Tagger* (Ver *Anexos B y C*). Como se podrá observar, se trata de un dispositivo informático programado en lenguaje Python que tiene como función principal la aplicación de las nociones propuestas en la metodología; esto con el fin de asistir al usuario en el etiquetado de tipo POS de manera manual de un corpus en una lengua que no disponga de los recursos para su automatización total, también de casos que requieran de un alto grado de intervención humana debido a la complejidad de la lengua o al nivel de exactitud requerido en el producto final.

Aunque este trabajo se escribe en lengua española, es necesario aclarar que la interfaz está diseñada para mostrar su información en inglés, con el fin de hacer la aplicación más accesible para investigadores de todo el mundo; puesto que esta lengua es más común en el medio académico.

6.1. Pasos iniciales

En la búsqueda de facilitar el manejo de la aplicación, uno de los pasos principales es la implementación de una interfaz de ventanas que represente visualmente con textos, botones y otros elementos, la información y las acciones que el usuario necesita conocer y puede realizar para lograr el etiquetado de su corpus. De esta manera, la primera interfaz que se presenta a la persona es una ventana que tiene las opciones necesarias para comenzar a etiquetar un nuevo corpus o recuperar el trabajo ya adelantado y comenzar en el punto en que se terminó en la ocasión anterior. A continuación, en la figura 2, se muestra esta primera ventana.

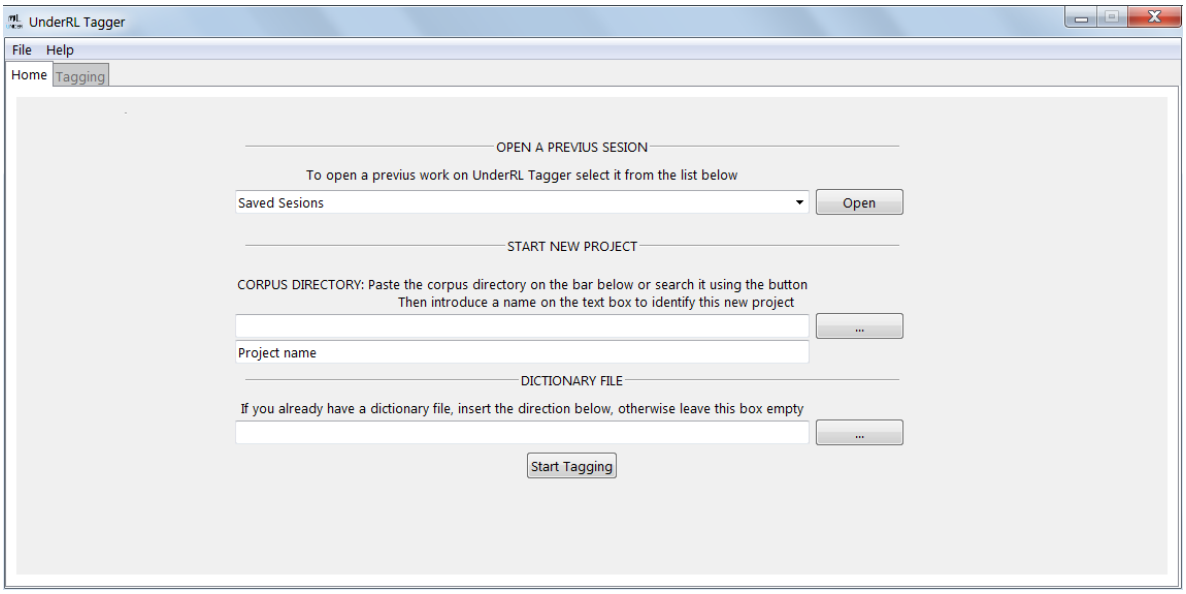

Figura 2. *Ventana del programa - Pantalla de inicio*

Como se mencionó anteriormente, el usuario tiene la posibilidad de elegir entre comenzar el etiquetado de un nuevo corpus o recuperar un trabajo previo. Por el momento, para comprender de mejor manera el funcionamiento de la aplicación, es necesario centrarse en lo que sucede cuando se inicia el etiquetado de un nuevo corpus. A continuación, se muestra lo que ocurre en la computadora cuando se selecciona esta opción por medio de un diagrama de flujo y luego se explican sus partes.

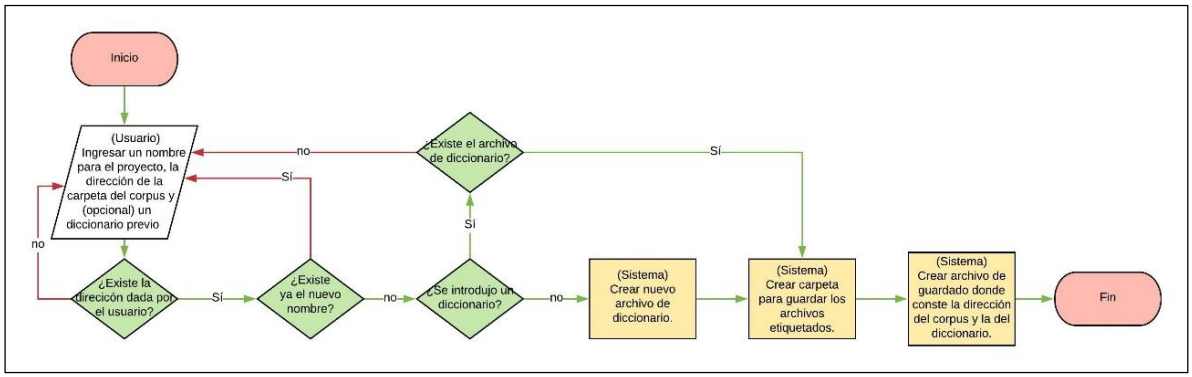

Figura 3. *Diagrama de flujo: Inicio de nuevo proyecto*

Tal y como lo muestra la figura 3, el objetivo de este primer proceso es la creación de las carpetas y los archivos necesarios para proceder al etiquetado. La obtención de la información sobre el directorio del corpus, su nombre y un posible archivo de diccionario previo que se desee reutilizar, se logra por medio de los campos de texto y botones que se encuentran bajo los títulos *START NEW PROJECT* y *DICTIONARY FILE* en la pantalla de

inicio; que permiten navegar por las carpetas del computador para encontrar la dirección y el archivo adecuados.

Una vez que el usuario pulsa el botón *Start tagging*, el programa realiza comprobaciones para verificar la existencia de la carpeta del corpus y del diccionario y para verificar si el nombre que se introdujo para el nuevo proyecto no está siendo usado por otro que se encuentre en la carpeta de *saved*. En caso de encontrar alguna inconsistencia con los datos introducidos por el usuario, el programa solicita, por medio de una ventana emergente, la corrección y no permite que el proceso continúe hasta que la verificación de los tres datos sea satisfactoria.

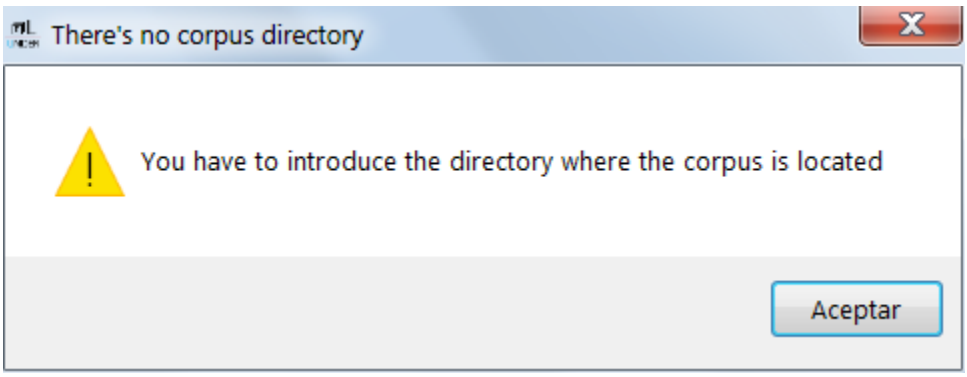

Figura 4. *Mensaje de advertencia*

Cuando se ha verificado la existencia de los datos en el equipo y la posibilidad de utilizar el nombre, la aplicación automáticamente crea una nueva carpeta en el mismo directorio del corpus con el nombre de *Tagged*; en ella se guardarán en adelante los archivos XML resultantes del proceso de etiquetado. Además de esta carpeta, en el caso de no haber un diccionario previo, se crea un nuevo archivo de diccionario que estará asociado al proyecto y finalmente se crea un archivo en la carpeta *saved* para llevar el control del avance en el proyecto; por el momento este archivo solamente tendrá la dirección de la carptea del corpus y la del diccionario, que por defecto se encontrará en la carpeta *dict*.

Los mencionados archivos de diccionario y de guardado se encontrarán en sus respectivas carpetas en el formato .txt o de texto plano, lo que permite una fácil lectura de ellos por el programa y también su fácil acceso para los usuarios que deseen, por ejemplo, reutilizar o compartir un diccionario ya creado.

También es muy importante tener en cuenta que por facilidad en el manejo de la información, y atentiendo a las recomendaciones propuestas por Parodi (2010) y citadas en el apartado *4.2.1* de este mismo trabajo, los textos que conformen el corpus deben encontrarse todos juntos en la misma carpeta y exclusivamente en el mismo formato de texto plano. Además de esto es necesario que hayan sido creados según la codificación UTF-8, que es un sistema que permite que el computador interprete de manera adecuada caracteres como letras con diferentes tipos de acentos. Esto es necesario para el correcto funcionamiento del programa y busca que se pueda reconocer un amplio conjunto de caracteres. Existen otros tipos de codificación utilizados para el reconocimiento de caracteres ajenos a los del alfabeto latino común a muchas lenguas europeas y americanas, pero su utilización se escapa a las intenciones de esta propuesta.

Por lo demás, las carpetas de *dict* y *saved* permanecerán siempre en la misma ubicación en que se ejecute el programa, de manera que un posible usuario que acceda a él obtendrá un archivo comprimido que contenga el archivo ejecutable del programa y estas dos carpetas previamente creadas, que no deberán ser movidas de la misma ubicación en ningún momento.

6.2. Inicio del etiquetado

En adelante, para ejemplificar el uso de la aplicación, se utilizará un pequeño texto conformado por una oración escrita en criollo sanandresano, una de las tantas URLa existentes en el territorio colombiano. En la siguiente tabla se muestra un breve análisis de esta, que será denominada oración A.

Tabla 2. *Descripción de Oración A*

ORACIÓN A							
Palabra	Di	bwai	gwain	da	di	niu	house
Categoría Gramatical	Artículo	Sustantivo	Verbo	Preposición	Artículo	Adjetivo	Sustantivo
Glosa	El	niño	va	a	la	nueva	casa

Teniendo esto en cuenta, una vez que se han recopilado los datos y creado los archivos necesarios, el programa habilita para el usuario la pantalla de etiquetado, que permite navegar entre los textos del corpus seleccionado, visualizar los avances realizados e interactuar con

las partes de un texto para añadirles las etiquetas correspondientes. A continuación, se muestra, en la figura 5, una vista de esta pantalla.

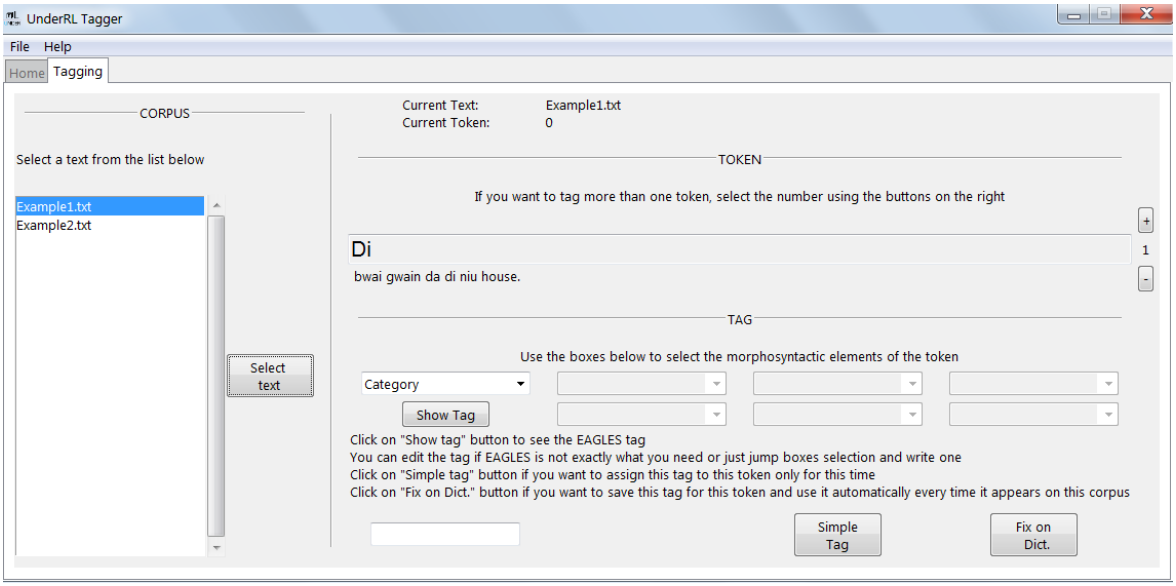

Figura 5. *Ventana del programa – Pantalla de etiquetado*

La pantalla de etiquetado se encuentra dividida en tres secciones principales, que se reconocen por la presencia de los títulos *CORPUS*, *TOKEN* y *TAG*. La primera de ellas permite visualizar una lista con los textos en formato .txt que el programa ha reconocido en la carpeta del corpus y seleccionar uno de ellos para etiquetarlo. Una vez elegido un texto, la sección *TOKEN* se activa, permitiendo visualizar el *token* que se encuentra seleccionado en el momento y varias unidades antes y después de él en el texto para proporcionar un contexto que permita al usuario reconocer la palabra según su función en el caso determinado. Así mismo, un par de botones, situados a la derecha de la barra que muestra el *token*, permiten seleccionar varias unidades para etiquetar unidades más grandes, una función muy útil en el caso de que se necesite etiquetar unidades pluriléxicas.

Continuando con el proceso, una vez que el programa permite visualizar el *token* seleccionado, también se activa la sección *TAG*, que permite asignar una etiqueta a la unidad mostrada. Para esta asignación se cuenta con un conjunto de menús desplegables con listas que varían según la categoría gramatical elegida, permitiendo seleccionar entre las diferentes posibilidades de análisis. Por ejemplo, en el caso de un sustantivo las listas permitirán asignar una etiqueta de número, género, nombre propio y grado; tal y como se muestra en la tabla 1.

Una vez que el usuario asigna las características del *token*, puede pulsar el botón *Show tag* para que la etiqueta sea traducida al estándar EAGLES y se muestre en la barra de texto de la parte inferior de la pantalla.

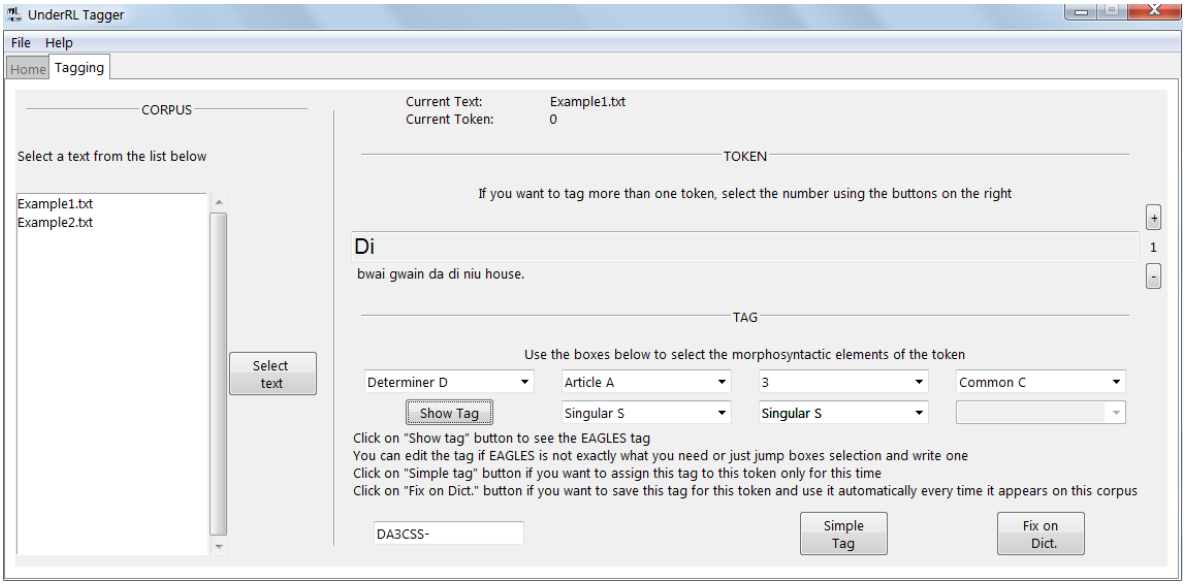

Figura 6. *Ejemplo de una unidad etiquetada con el sistema de listas desplegables*

La etiqueta que aparece en la parte inferior, sin embargo, no es definitiva; el usuario puede editarla según desee para complementar con información ajena a la proporcionada por el estándar EAGLES, o de la manera en que lo exijan los intereses de su propia investigación y la información que considere importante tener sobre las unidades. Esto permite, además, que el usuario pueda introducir en el mismo campo una etiqueta completamente propia, sin pasar por las listas desplegables, en caso de que desee ceñirse a otro estándar o tenga su manera de codificar las etiquetas.

Sin embargo, para que la utilización de las secciones *TOKEN* y *TAG* sea posible, es necesario procesar el texto de una manera específica. Este proceso se lleva a cabo en el momento en que el usuario pulsa el botón *Select text* y se muestra a continuación en la figura 7.

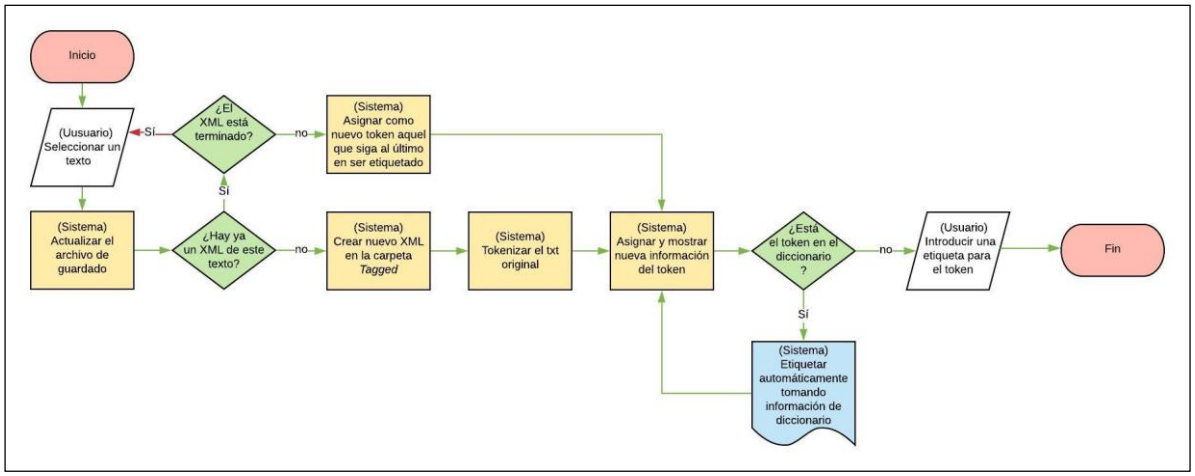

Figura 7. *Diagrama de flujo: Procesamiento previo de un texto seleccionado*

Como puede observarse en el diagrama, al seleccionar un texto, el programa comprueba si ya hay un avance en su etiquetado o si es necesario comenzar desde el primer *token*; después de esta evaluación, el sistema tokeniza el texto.

El proceso de tokenizado, siguiendo a Mitkov (2004), puede entenderse como la división del texto original en componentes mínimos, secuencias de caracteres que se corresponden con unidades básicas lingüísticas como la palabra, la puntuación o los números. A cada una de las unidades resultantes se le denomina *token* (p. 210).

Es importante tener en cuenta que hay una diferencia entre los conceptos de *token* o *token word* y palabra, en primera instancia porque, en la práctica, el tokenizado no separa solamente cada unidad léxica del texto, sino que también se consideran como *tokens* las unidades de puntuación como el punto, la coma, los signos de interrogación y admiración, etc. así como las cantidades, fechas, porcentajes y otras apariciones de caracteres numéricos. Por otra parte, el concepto de palabra, entendido como sinónimo de lexema, recoge una unidad de forma y significado que es única independientemente de sus apariciones múltiples, mientras que el de *token* se refiere a cada serie de caracteres por separado, por lo que cada aparición de una misma palabra en el texto será un *token* diferente, identificado por el sistema de manera única frente a todos los demás del mismo conjunto (p. 211).

Así mismo, teniendo en cuenta que una de las maneras de reconocer el *token* es por medio de la separación de un espacio en blanco que media entre la mayoría de las palabras, se hace necesario reconocer que existen algunas unidades que pueden componer una unidad

léxica y llevar un espacio en blanco entre sus componentes. Ejemplos de ello son las locuciones o los números en lenguas como el francés y el español. Este tipo de unidades se denomina *multi-token word* (p. 212) y es necesario brindar la posibilidad al usuario de seleccionar varios *tokens* con el objetivo de poderlas conformar según sea necesario para su proyecto de etiquetado; de ahí que esta aplicación permita la selección de varios *tokens*.

Una vez que el texto se encuentra tokenizado, el programa guarda cada uno de los *tokens* que lo componen en una lista ordenada, en la que cada uno cuenta con un número que lo identifica individualmente dentro de todo el conjunto. Teniendo esta información, la aplicación selecciona el *token* que se mostrará en la pantalla; en el caso de un texto ya iniciado, se muestra el primer *token* que se haya dejado sin etiquetar en la sesión previa y en caso de ser un texto completamente nuevo, se selecciona y muestra el primer *token* de la lista.

Teniendo la información del *token* seleccionado, el programa realiza una búsqueda en el diccionario para verificar si ya fue etiquetado antes y puede replicar esa etiqueta en este caso, de esta manera se automatiza parte del proceso. Por el momento, si se asume que es un proyecto nuevo con un diccionario vacío, el programa solamente mostrará el primer *token* del texto, tal y como se ve en la figura 5 y permitirá al usuario elegir la etiqueta deseada para él.

6.3.Etiquetas, salidas y diccionario

En el momento en que se ha seleccionado un texto y se cuenta con los archivos y carpetas necesarios, el programa permitirá asignar una etiqueta a cada uno de sus *tokens*. Una vez que la etiqueta se ha introducido en la barra inferior, tal y como se muestra en la figura 6, el usuario tiene dos maneras de asignarla al *token*: la primera de ellas se realiza utilizando el botón *Simple tag*, que permite asociar esa etiqueta a esa única aparición del *token*; la segunda opción se ejecuta utilizando el botón *Fix on Dict*., que permite fijar una entrada en el diccionario con ese *token* asociado a esa etiqueta. En ambos casos, la etiqueta determinada será escrita en el archivo XML de salida correspondiente, junto con el *token* actual y su número de identificación dentro del texto. Este proceso se muestra a continuación en la figura 8.

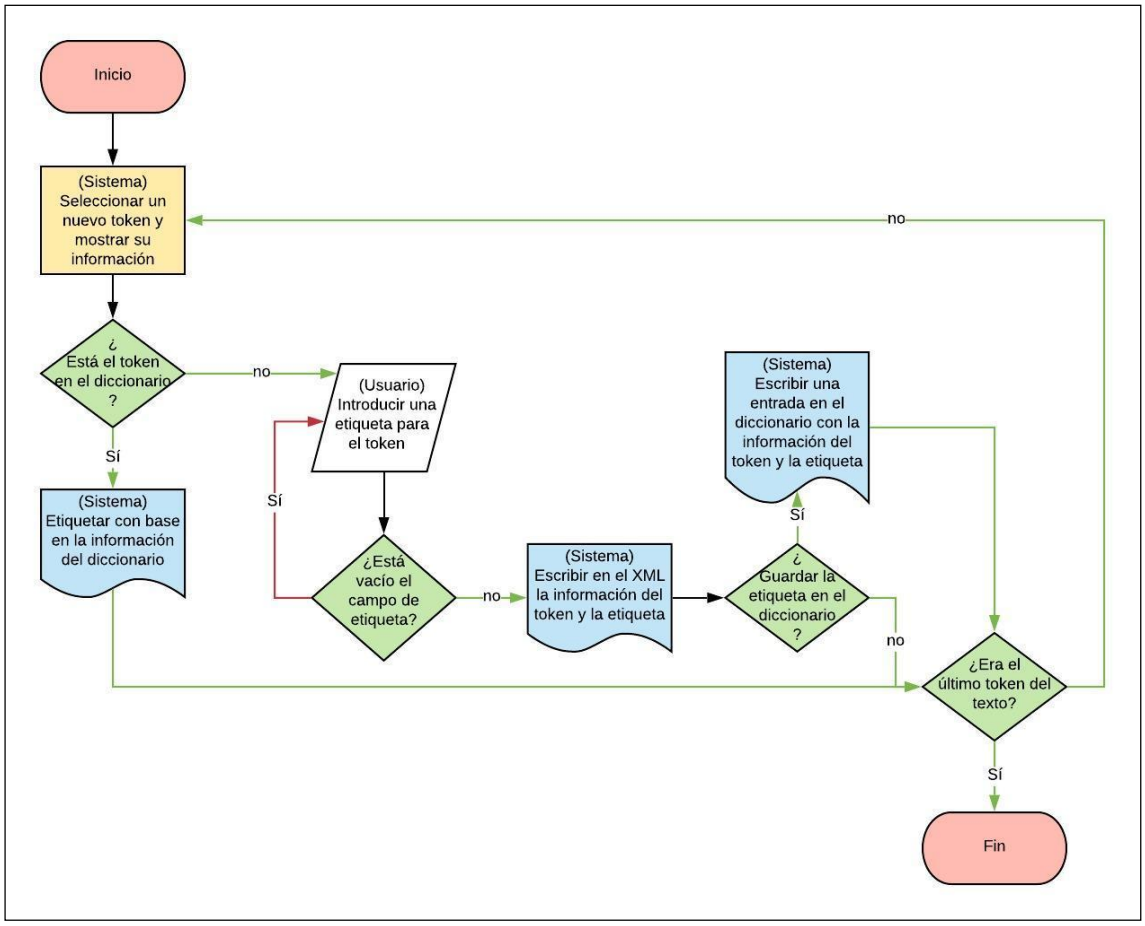

Figura 8. *Diagrama de flujo: Etiquetado y escritura en XML*

De la misma manera que en el proceso del numeral 6.2, una vez que se selecciona un *token*, la primera comprobación que se realiza es buscarlo en el diccionario, lo que llevaría a su etiquetado automático. Sin embargo, suponiendo nuevamente que se trata de un proyecto nuevo con un diccionario vacío, el programa permitirá que el usuario introduzca una etiqueta para el *token* de manera manual. Una vez que se pulsa cualquiera de los dos botones (*Simple Tag* o *Fix on Dict*.), es necesario comprobar que no se encuentra vacía la casilla de la etiqueta, con el fin de evitar que se desencadenen errores como la presencia de *tokens* sin etiquetar o la escritura de entradas vacías en el diccionario. En caso de que el usuario desee dejar un *token* con una etiqueta vacía, puede seleccionar algún carácter especial que juegue este papel dentro de su sistema de etiquetas, por ejemplo, una *X*, un punto, un *0* o un guion bajo.

Cuando se ha introducido una etiqueta y se ha pulsado uno de los dos botones mencionados, la aplicación escribirá, en ambos casos, la información necesaria en el documento

de salida, es decir, el documento XML que corresponde al texto que se encuentre seleccionado. Esta información consiste en una línea de texto que se añade a las que estén escritas previamente, de manera parecida a lo mostrado en la figura 1. Se muestra a continuación un ejemplo en que se aplica este proceso a la oración A.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<text name="Ejemplo1.txt">

<token form="Di" tag="DA3CNS-" id="t.0.1"/>
<token form="bwai" tag="NCMS—" id="t.1.1"/>
<token form="gwain" tag="VMIP3SC" id="t.2.1"/>
<token form="da" tag="SP——" id="t.3.1"/>
<token form="di" tag="DA3CNS-" id="t.4.1"/>
<token form="niu" tag="AQ-FS-S" id="t.5.1"/>
<token form="house" tag="NCFS—" id="t.6.1"/>
<token form="." tag="Fp" id="t.7.1"/>
</text>
```

Figura 9. *Oración A etiquetada en XML*

En esta muestra pueden apreciarse los elementos que tendrán todos los archivos XML de salida etiquetados con esta herramienta. En primer lugar, se cuenta con un encabezado, en la primera línea, que hace parte de la gramática de este lenguaje y cuenta con información importante para que el archivo sea interpretado por el ordenador. Luego de esto está el elemento *text*, que tiene como atributo el nombre del texto actual y que contiene dentro de sí todos los *tokens* etiquetados.

La parte principal del archivo está compuesta por cada uno de los diferentes *tokens* que conforman el texto. Cada uno de ellos es un elemento de tipo *token* que cuenta con tres atributos: *form*, *tag* e *id*; el primero de ellos contiene la cadena de caracteres que pertenece al *token*, es decir, la palabra, el número o el signo que estamos etiquetando; el segundo contiene la etiqueta que le fue asignada y el tercero contiene el número que lo identifica. Este número está compuesto por la letra *t*, un primer número, que incrementa de uno en uno con cada *token* y un segundo número, que dice cuántas unidades componen el *token*; en el caso de una *multi-token word*, este número varía según los elementos que la compongan. La línea del final indica que hasta ese punto llega el elemento *text* y que su contenido es todo lo que se encuentra antes de ella.

Con la información escrita en el XML, si el usuario seleccionó el botón *Fix on Dict.*, se procederá a escribir una nueva entrada en el diccionario, que permita recuperar esa misma etiqueta para ese mismo *token* tantas veces como este último aparezca en el corpus, permitiendo así que la acción de etiquetado se realice automáticamente sin que el usuario deba introducir de nuevo las mismas características para la unidad.

Esta última opción es útil para unidades que sabemos que aparecerán siempre con la misma etiqueta POS, tales como, por ejemplo, los signos de puntuación y otras expresiones de aparición constante en una lengua, como adverbios o preposiciones. Incluso, algunos *tokens* que se identifiquen como verbos, sustantivos o adjetivos, pueden etiquetarse de esta manera siempre que se tenga seguridad sobre la repetición en los resultados de su análisis. En la oración A, por ejemplo, puede identificarse la palabra *Di*, que invariablemente representará un artículo determinado que es invariable tanto en el número como en el género, puesto que se comporta de manera parecida al *the* del inglés. En este caso, entonces, podría crearse, con toda confianza, una entrada en el diccionario para este *token*. Procesos parecidos podrían seguirse con otras partes de la misma oración, por lo que el *Simple tag* sería necesario solamente en los casos en que el mismo *token* sea susceptible de diferentes análisis dependiendo de su contexto, su función o de la homografía entre palabras. A continuación, se muestra un archivo de diccionario realizado a partir de la oración mencionada.

```
entry_ . ***** Fp
entry_ bwai ***** NCMS---
entry_ di ***** DA-CNS-
entry_ house ***** NCFS---
entry_ niu ***** AQ-CS--
```

Figura 10. *Ejemplo de entradas en diccionario*

Cada una de las líneas de este documento es una entrada, que contiene un *token* y su etiqueta correspondiente. Además de estos dos elementos, la entrada contiene los caracteres *entry_* al inicio y ***** entre el *token* y la etiqueta. La utilización de estos componentes extra se debe a que es necesario tener una manera de delimitar la extensión del *token*, puesto que, como se mencionó anteriormente, la simple existencia de espacios en blanco al principio y al final no es suficientemente infalible a causa de la existencia de las *multi-token word*. Esto significa

que cada vez que el programa desea obtener la información de una entrada, está programado para saber que el *token* es todo aquello que se encuentra entre *entry_* y *****, independientemente de los espacios, la puntuación o los caracteres especiales que pueda haber; así mismo, identifica la etiqueta como todo aquello que se encuentra en la misma línea después de *****.

Por otra parte, también es notorio en la figura 10 el hecho de que las entradas se encuentran organizadas alfabéticamente según el *token*. Esto se debe a que cada vez que el programa introduce una nueva, reorganiza todo el documento buscando este orden, pues de esta manera la tarea de búsqueda se lleva a cabo de forma más rápida y eficiente (Cormen, 2013, p. 25); además, esta forma de proceder facilita también la búsqueda manual por parte del usuario, lo que puede ser útil en el caso de que se desee enmendar un error en una entrada.

Al terminar el etiquetado del *token*, el sistema comprobará que no se haya tratado del último que componía el texto; si es así, se avisará al usuario de la finalización del texto y se le solicitará que seleccione otro, de lo contrario, se continuará con el siguiente *token* en la lista, realizando con él todo el proceso desde el comienzo.

6.4. Procesamiento automático

Continuando con el mismo ejemplo, se añadirá ahora un segundo texto con una oración diferente en la misma lengua, denominada Oración B, que cuenta con los elementos que se muestran a continuación:

Tabla 3. *Descripción de Oración B*

ORACIÓN B							
Palabra	Di	bwai	plie	wid	di	niu	baal
Categoría Gramatical	Artículo	Sustantivo	Verbo	Preposición	Artículo	Adjetivo	Sustantivo
Glosa	El	niño	juega	con	la	nueva	pelota

Como puede verse, las oraciones A y B comparten algunas palabras, lo que se traduce en que también comparten la información ligada a algunos de sus *tokens*; estos son: *di*, *bwai* y *niu*. Suponiendo que al etiquetar la Oración A fueron introducidas en el diccionario las entradas

que se muestran en la figura 10, el programa cuenta ya con la información necesaria para automatizar una parte del proceso de etiquetado de la Oración B.

Para comprender el funcionamiento de esta automatización, es necesario volver sobre la figura 8, en la que se muestra que el primer proceso que realiza la aplicación ante un nuevo *token* es su búsqueda en el diccionario. Si se llega, entonces, al archivo de texto que contiene la Oración B, el programa tomará *Di*, el primer *token*, y lo buscará en el diccionario.

Tratándose del mismo diccionario de la figura 10, el *token* será encontrado y se escribirá automáticamente, sin intervención del usuario, la etiqueta correspondiente en el archivo XML de salida para la nueva oración. Lo mismo sucederá con el *token* siguiente: *bwai*, pues su información también fue introducida durante el etiquetado de la Oración A. De esta manera, la herramienta solamente pasará a mostrar al usuario aquel *token* que no se halle en el diccionario y que requiera, por ello, de un etiquetado manual. En este caso, se trata del *token plie*, como se muestra en la siguiente figura:

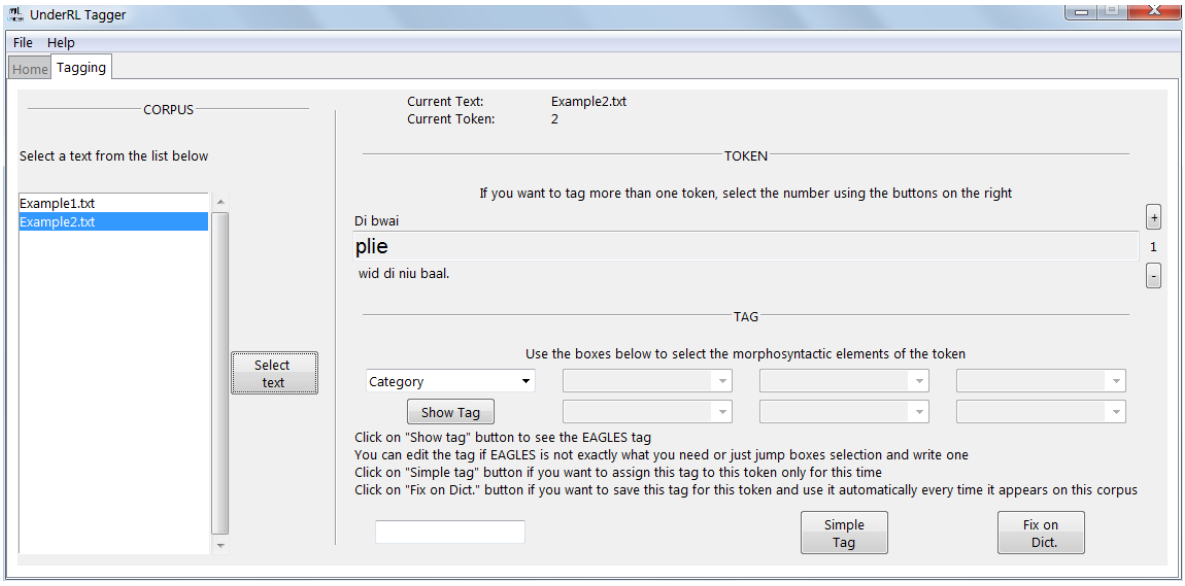

Figura 11. *Pantalla de etiquetado después de automatizar los primeros tokens*

Así mismo, una vez que sean etiquetados de manera manual los *tokens plie* y *wid*, se pasará a *di* y *niu*, que serán también etiquetados automáticamente utilizando el diccionario; de esta manera, el programa siempre realiza la búsqueda cada vez que se pasa a un nuevo *token*, con el fin de automatizar el proceso en la medida de lo posible, sea cual fuere el punto del texto en que se encuentre.

La búsqueda en el diccionario, sin embargo, no funciona solamente con la forma más sencilla del *token*, sino que también debe tener en cuenta la presencia de las *multi-token word* que el usuario haya guardado. A continuación, se muestra de manera más detallada el proceso de búsqueda en el diccionario:

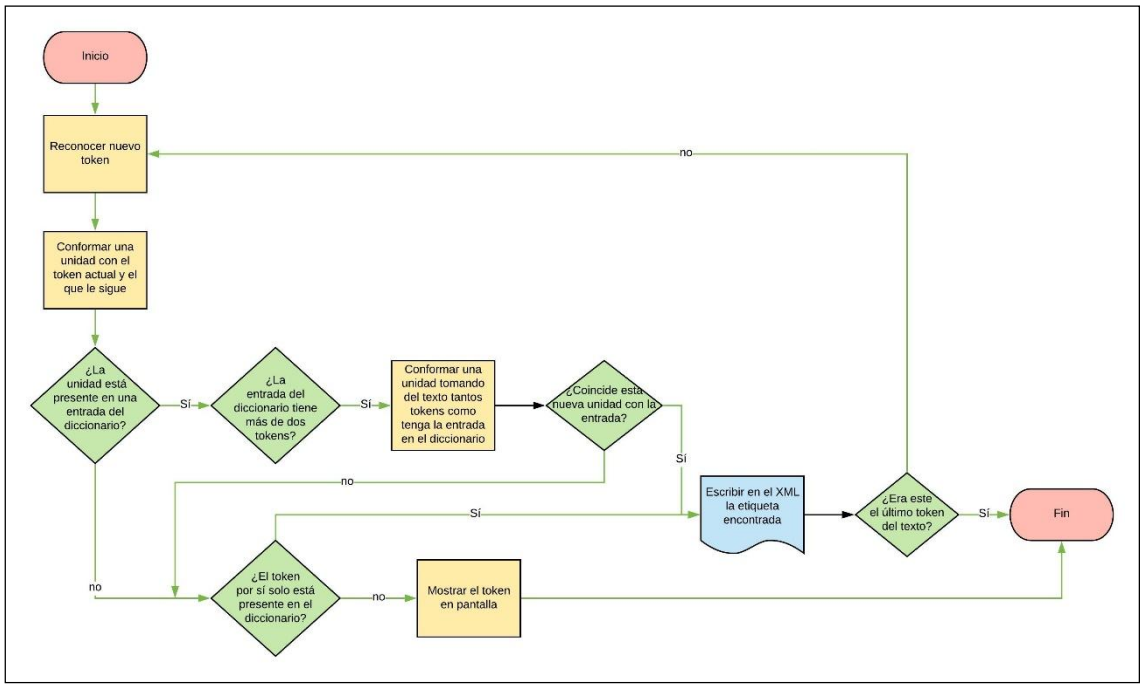

Figura 12. *Diagrama de flujo: Búsqueda en diccionario*

Tal y como lo muestra la figura 12, cuando el sistema se dispone a comprobar la existencia de un *token* determinado en el diccionario, en realidad comienza por buscar si hay una *multi-token word* asociada a él. Para esto, crea una unidad compuesta por el *token* en cuestión y aquel que le sigue en el texto y es esta unidad la que compara con las entradas del diccionario.

Si dicha unidad compuesta es encontrada como parte de alguna de las entradas, se verifica el número de *tokens* que componen la entrada, con el fin de comprobar si se compone de dos o más *tokens*. En caso de que se componga solo de dos, estos habrán coincidido ya de manera exacta con aquellos que se tomaron del texto, es decir, con la unidad creada al principio, por lo que no se requieren más comprobaciones para asumir que el *token* original hace parte de una *multi-token word* y se puede utilizar la etiqueta encontrada para escribirla en el archivo de salida.

En el caso de que la entrada del diccionario contenga a la unidad creada, pero posea más de dos *tokens*, no será posible asumir que se trate de la misma y se hará necesario tomar más *tokens* del texto para comprobarlo. Suponiendo, por ejemplo, que se toma un *token A* y

un *token B* para crear la unidad compuesta y que la entrada del diccionario se compone de un *token A*, un *token B* y un *token C*; sería necesario comprobar que el *token* que sigue al *token B* en el texto es el mismo que aquel que lo sigue en el diccionario. Esto se logra creando una nueva unidad conformada por *tokens* del texto en un número igual al que posee la entrada. En caso de que ambas unidades resulten ser iguales, podrá asumirse entonces que hay una *multi-token word* y se escribirá su etiqueta en el archivo XML de salida. Tanto en este caso como en el anterior, el programa se saltará los *tokens* que hagan parte de la unidad y continuará con el etiquetado en el que la sigue inmediatamente.

Por otra parte, si ninguna de estas dos tentativas de encontrar que el *token* es parte de una *multi-token word* tiene resultado positivo, entonces el sistema procederá a buscar el *token* por sí solo en el diccionario, es decir que buscará una entrada que lo contenga de manera exacta, sin otros *tokens* alrededor. De encontrarla, igualmente se escribirá su información en el archivo de salida y se pasará al siguiente *token* del texto para realizar las comprobaciones necesarias.

Repitiendo este mismo proceso durante varios textos y estando atento a las unidades cuya información puede reutilizarse sin temor a cometer errores, un investigador puede conformar, en poco tiempo, un diccionario de base que le sirva para agilizar el proceso de etiquetado de POS en una lengua determinada, que además se irá robusteciendo a medida que se encuentren y agreguen más entradas. Al mismo tiempo, el programa solicitará información nueva cada vez que sea necesario y permitirá que todos los archivos asociados al corpus sean editados manualmente en caso de que sea necesario enmendar errores.

Así, gracias a los procesos descritos en este capítulo, se puede establecer una manera de automatizar, en gran medida, el proceso de etiquetado de un corpus de textos en el caso de que no se cuente con los recursos informáticos para que la tarea se realice con una mínima intervención humana.

7. Conclusiones y perspectivas

En las páginas anteriores ha sido posible observar que efectivamente es posible, a través de un conjunto de algoritmos sencillos, crear una herramienta informática que tenga como objetivo la automatización de varios de los pasos necesarios para el etiquetado manual de POS en URLa. Tal solución consiste en el manejo de diferentes tipos de archivo, siendo el más importante un diccionario de etiquetas que permite la reutilización de la información introducida previamente por un investigador. A su vez, se destaca también la implementación de una interfaz gráfica de ventanas que permite la comunicación entre los diferentes archivos, el corpus de textos y el usuario.

Para construir esta herramienta ha sido importante, además, explorar la relación que hay entre el PLN y la lingüística aplicada, así como las aplicaciones y utilidades que puede tener el primero en los problemas propios de la lingüística de corpus, como el manejo de grandes cantidades de datos o la velocidad en el procesamiento de la información. Así mismo, ha sido necesario contar con las definiciones que atañen al concepto de *under-resourced languages* y ver que una aplicación como la propuesta en los apartados anteriores podría ayudar a constituir los primeros pasos en la consecución de unos medios más inclusivos, que permitan la aplicación de las ventajas de la era digital a medios con hablantes de estas lenguas, así como su investigación y desarrollo en términos académicos.

En función de este objetivo estuvo la construcción de un sistema de etiquetas basado en las propuestas de EAGLES y en el lenguaje XML, así como el reconocimiento de que los diferentes casos de estudio pueden requerir de unas etiquetas diferentes; en este sentido, se ha construido la propuesta con la intención de que haya libertad en la selección y fijación de las etiquetas, permitiendo que el investigador mismo sea quien asuma una teoría lingüística subyacente a su etiquetado que se ajuste a los objetivos y metas de sus posibles trabajos.

Por otra parte, también se ha construido un formato de diccionario en el que puedan constar los diferentes *tokens* y su información en lo que se refiere al POS, con el fin de que pueda ser compartido y reutilizado por diferentes investigadores y grupos alrededor del mundo que tengan la necesidad de trabajar con las URLa. Este sistema de construcción y lectura del diccionario está pensado para que sea fácil de corregir, editar y compartir según las necesidades del usuario, pero manteniendo siempre unas convenciones que permitirán a la herramienta interpretar adecuadamente la información allí contenida.

Los algoritmos descritos a través de los diagramas de flujo y su funcionalidad son, entonces, la conclusión más importante de este trabajo. Sin embargo, se plantea además, como perspectiva, la posibilidad de compartir, con la ayuda de los medios informáticos de que disponen la Universidad de Antioquia y del semillero *Corpus Ex Machina*, una pieza de software de libre acceso construida con estos parámetros, que esté al servicio de la comunidad académica.

Así mismo, se espera construir en un futuro, apoyada por la misma institución y el semillero, una plataforma en la que sea posible compartir los avances alcanzados por los investigadores en las diferentes lenguas, de manera que su trabajo pueda ser reutilizado por otros investigadores y así se logre que el etiquetado de las URLa pueda ser automatizado en la medida de lo posible. Esta propuesta permitirá avances en el etiquetado y la construcción de corpus, y, además, ayudará al fortalecimiento de una comunidad dispuesta al trabajo colaborativo en pro de las necesidades de comunidades lingüísticas hasta ahora con poco acceso a los recursos que ofrecen las aplicaciones contemporáneas de las ciencias del lenguaje.

8. Bibliografía

Anthony, L. (2015). *TagAnt (Version 1.2. 0)[Computer Software]*. Waseda University. http://www.laurenceanthony.net/software/tagant/

Baquero, J. M. (2010). *Lingüística computacional aplicada*. Universidad Nacional de Colombia.

Berment, V. (2004). *Méthodes pour informatiser les langues et les groupes de langues « peu dotées »* [Tesis doctoral, Université Joseph-Fourier - Grenoble I]. https://tel.archives-ouvertes.fr/tel-00006313

Bernal, J., & Hincapié, D. (2018). *Lingüística de corpus*. Instituto Caro y Cuervo. https://www.academia.edu/36731349/Ling%C3%BC%C3%ADstica_de_corpus

Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, *56*, 85–100.

Brezina, V. (2018). *Statistics in corpus linguistics: A practical guide*. Cambridge University Press.

Buseman, K., & Buseman, A. (2013). *Field Linguist's ToolBox (Version 1.6.1)*. SIL International. https://software.sil.org/toolbox/

Cormen, T. H. (2013). *Algorithms Unlocked*. MIT Press.

El-Haj, M., Kruschwitz, U., & Fox, C. (2015). Creating language resources for under-resourced languages: Methodologies, and experiments with Arabic. *Language Resources and Evaluation*, *49*(3), 549–580.

González, M. S., & Rodríguez, M. L. (2010). *Lenguas indígenas de Colombia: Una visión descriptiva*. Instituto Caro y Cuervo.

Hausser, R. (2013). *Foundations of Computational Linguistics: Human-Computer Communication in Natural Language*. Springer Science & Business Media.

Krauwer, S. (2003). The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. *Proceedings of SPECOM 2003*, 8–15.

Le, V.-B., & Besacier, L. (2009). Automatic speech recognition for under-resourced languages: Application to Vietnamese language. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(8), 1471–1482.

Leech, G., & Wilson, A. (1996). *EAGLES recommendations for the morphosyntactic annotation of corpora*. Istituto di Linguistica Computazionale. http://www.ilc.cnr.it/EAGLES96/annotate/node1.html

Maxwell, M., & Hughes, B. (2006). Frontiers in linguistic annotation for lower-density languages. *Proceedings of the workshop on frontiers in linguistically annotated corpora 2006*, 29–37.

McEnery, T., & Hardie, A. (2011). *Corpus Linguistics*. Edinburgh University Press.

McEnery, T., & Hardie, A. (2013). The history of corpus linguistics. *The Oxford handbook of the history of linguistics*, *727*, 745.

Mitkov, R. (2004). *The Oxford Handbook of Computational Linguistics*. OUP Oxford.

Moe, R. (2008). FieldWorks Language Explorer 1.0. *SIL Forum for Language Fieldwork 2008-011*. SIL Forum for Language. https://www.sil.org/system/files/reapdata/16/65/64/166564652392804523031385956657080191552/SIL-Forum2008_011.pdf

Molina Mejia, J. M., González-Rátiva, M. C., Pemberty Tamayo, J. L., Grajales Ramírez, A. F., & Bermúdez Cardona, A. (2017, mayo). Hacia la anotación y etiquetado de un corpus sociolingüístico: Preseea-Medellín. *Congreso Internacional de Lingüística Computacional y de Corpus*. https://halshs.archives-ouvertes.fr/halshs-02012701

Molina Mejía, J. M., Grajales Ramírez, A. F., & Pemberty Tamayo, J. L. (2019). Hacia un dispositivo informático basado en Corpus para la Enseñanza del Español Lengua Extranjera (DICEELE). *Revista Internacional de Tecnología, Conocimiento y Sociedad*, *7*(1), 1–13.

Moreno Sandoval, A. (1998). *Lingüística computacional: Introducción a los modelos simbólicos, estadísticos y biológicos*. Editorial Síntesis.

Nerbonne, J. (2007). Linguistic Challenges for Computationalists. En N. Nicolov, *Recent Advances in Natural Language Processing IV. Selected papers from RANLP 2005* (pp. 1–16). John Benjamins Publishing.

Padró, L., Collado, M., Reese, S., Lloberes, M., & Castellón, I. (2010). Freeling 2.1: Five years of open-source language processing tools. *7th International Conference on Language Resources and Evaluation*.

Parodi, G. (2010). *Lingüística de corpus: De la teoría a la empiria*. Iberoamericana.

Rogers, C. (2010). Review of fieldworks language explorer (flex) 3.0. *Language Documentation & Conservation*, *4*, 78–84.

Sáiz Noeda, M. (2002). Influencia y aplicación de papeles sintácticos e información semántica en la resolución de la anáfora pronominal en español. *Procesamiento del lenguaje natural*, *28*, 113–114.

Scannell, K. P. (2007). The Crúbadán Project: Corpus building for under-resourced languages. *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, *4*, 5–15.

Schmid, H. (1994). *TreeTagger-a language independent part-of-speech tagger*. https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Sinclair, J. M. (1992). The automatic analysis of corpora. En *Directions in Corpus Linguistics*. Mouton de Gruyter, Berlin and New York.

Tognini-Bonelli, E. (2001). *Corpus Linguistics at Work*. John Benjamins Publishing.

Tordera Yllescas, J. C. (2011). *Lingüística computacional: Tecnologías del habla*. Publicacions de la Universitat de València.

Weisser, M. (2018, octubre 28). *Automatically Enhancing Tagging Accuracy and Readability for Common Freeware Taggers*. Asia Pacific Corpus Linguistics (APCLC) 2018, Takamatsu, Japón.

9. Anexos

Anexo A: Lista completa de las etiquetas utilizadas por el programa

La tabla a continuación muestra los diferentes valores y caracteres que el programa puede utilizar por defecto para construir las etiquetas de los *tokens*; a su vez, estas se basan en las usadas por las del sistema *Freeling*[15] (Padró et al., 2010) siguiendo los lineamientos de EAGLES (Leech & Wilson, 1996).

Categoría Gramatical	Característica	Posibles valores		
A Adjective	Type	O (Ordinal)	Q (Qualificative)	P (Possessive)
	Degree	S (Superlative)	V (Evaluative)	
	Genere	F (Feminine)	M (Masculine)	C (Common)
	Number	S (Singular)	P (Plural)	N (Invariable)
	Possessor person	1	2	3
	Possessor number	S (Singular)	P (Plural)	N (Invariable)
C Conjunction	Type	C (Coordinating)	S (Subordinating)	
D Determiner	Type	A (Article)	D (Demonstrative)	I (Indefinite)
		P (Possessive)	T (Interrogative)	E (Exclamative)
	Person	1	2	3
	Genere	F (Feminine)	M (Masculine)	C (Common)
	Number	S (Singular)	P (Plural)	N (Invariable)
	Possessor number	S (Singular)	P (Plural)	N (Invariable)

[15] Las EAGLES usadas por *Freeeling* pueden encontrarse en su página web de documentación: https://talp-upc.gitbook.io/freeling-4-0-user-manual/tagsets/tagset-es.

N **Noun**	Type	C (Common)	P (Proper)	
	Genere	F (Feminine)	M (Masculine)	C (Common)
	Number	S (Singular)	P (Plural)	N (Invariable)
	Neclass	S (Person)	G (Location)	O (Organization
		V (Other)		
	Degree	V (Evaluative)		
P **Pronoun**	Type	D (Demonstrative)	E (Exclamative)	I (Indefinite)
		P (Personal)	R (Relative)	T (Interrogative)
	Person	1	2	3
	Genere	F (Feminine)	M (Masculine)	C (Common)
	Number	S (Singular)	P (Plural)	N (Invariable)
	Case	N (Nominative)	A (Accusative)	D (Dative)
		Q (Oblique)		
	Polite	P (Yes)		
R **Adverb**	Type	N (Negative)	G (General)	
S **Adposition**	Type	P (Preposition)	S (Postposition)	C (Circumposition)
		Z (Particle)		
V **Verb**	Type	M (Main)	A (Auxiliary)	S (Semiauxiliary)
	Mood	I (Indicative)	S (Subjunctive)	M (Imperative)
		P (Participle)	G (Gerund)	N (Infinitive)
	Tense	P (Present)	I (Imperfect)	F (Future)
		S (Past)	C (Conditional)	

	Person	1	2	3
	Number	S (Singular)	P (Plural)	N (Invariable)
	Genere	F (Feminine)	M (Masculine)	C (Common)
Z **Number**	Type	d (Partitive)	m (Currency)	p (Percentage)
		u (Unit)		
W **Date**				
I **Interjection**				
Punctua-tion	Type	Fp (Period)	Fc (Comma)	FD (Colon)
		Fx (Semicolon)	Fit (Questionmark close)	Fia (Questionmark open)
		Fat (Exclamationmark close)	Faa (Exclamationmark open)	Fpt (Parenthesis close)
		Fpa (Parenthesis open)	Fe (Quotation)	Frc (Quotation close)
		Fra (Quotation open)	Flt (Curlybracket close)	Fla (Curlybracket open)
		Fs (Suspension points)	Fg (Hyphen)	Fz (Other)
		Ft (Percentage)	Fh (Slash)	Fct (Squarebracket close)
		Fca (Squarebracket open)		

Anexo B: Archivo ejecutable de *UnderRL Tagger*

Como anexo B, se entrega con este trabajo un archivo comprimido (.rar) que contiene un archivo ejecutable para el sistema Windows (.exe) del programa que se describe en el cuerpo

del trabajo, así como las carpetas para su adecuado funcionamiento según también se explicó en las páginas anteriores.

Anexo C: Archivo en lenguaje Python de *UnderRL Tagger*

Como anexo C, también se entrega un archivo en lenguaje Python (.py), que constituye el código fuente del programa, con todos los algoritmos que fueron descritos anteriormente y que puede ser utilizado para ejecutarse en cualquier sistema operativo siempre que se cuente con un intérprete de dicho lenguaje.