LUZ MARINA SIERRA MARTÍNEZ

Algoritmo Memético para la Identificación de Partes del Discurso

Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para la obtención del Título de

Doctora en:
Ingeniería Telemática

Director:
PhD. Juan Carlos Corrales Muñoz

Codirector:
PhD. Carlos Alberto Cobos Lozada

Asesor:
PhD. Tulio Rojas Curieux

Popayán
2018

Página de aceptación

_____

_____

_____

_____
Ph.D. Juan Carlos Corrales Muñoz
Director -Tutor

_____
Ph.D. Carlos Alberto Cobos Lozada
Codirector

_____
Presidente del Jurado

_____
Jurado

_____
Jurado

Popayán, 10 de mayo de 2018

A mi familia

Agradecimientos

Al finalizar mi tesis de doctorado, es indispensable expresar mis sentimientos de agradecimiento hacia todas las personas e instituciones que me han brindado su apoyo y colaboración para alcanzar esta anhelada meta.

Doy gracias a Dios, por brindarme las fuerzas, la salud, y por todas las bendiciones requeridas para finalizar mi objetivo.

Deseo agradecer a mis Directores de Tesis y Tutores, Juan Carlos Corrales y Carlos Alberto Cobos, por brindarme la oportunidad de aprender a su lado, de guiarme en todos los momentos durante la ejecución de este proyecto y del desarrollo del doctorado. Les doy gracias por todo. De igual manera, le agradezco al profesor Tulio Rojas, por brindarme su apoyo incondicional y sus conocimientos sobre la lengua nasa, que han sido tan importantes para alcanzar tan anhelado objetivo. Gracias, a todos por abrirme la puerta del conocimiento.

Muchas gracias a los profesores Enrique Herrera Viedma y Diego Peluffo Ordoñez por su apoyo para realizar mis estancias de investigación y por haberme acogido como un miembro más de sus equipos de trabajo.

Agradezco a todos los amigos y amigas, por el apoyo que me han brindado en todo momento durante el desarrollo de este doctorado.

Agradezco también al programa de Doctorado en Ingeniería Telemática, por haberme aceptado y apoyado como estudiante en tan reconocido programa. De igual manera, agradezco al Departamento de Sistemas, a las directivas de la FIET y de la Universidad del Cauca, por haberme brindado su apoyo para realizar mis estudios de doctorado, especialmente por otorgarme la comisión de estudios para alcanzar esta meta propuesta.

He dejado para el final, para expresar mi profundo agradecimiento a mi familia, por todo el cariño, amor y colaboración que siempre me han brindado, lo cual me ha mantenido firme y me ha dado la fuerza para seguir adelante. Mil gracias, a mi madre

por sus consejos, oraciones y apoyo incondicional. A Oscar, Jordín, Luis y Muñeco, por todo el afecto, amor y apoyo que siempre he recibido.

¡¡¡Gracias!!!

Resumen estructurado

El crecimiento de las comunicaciones a través de dispositivos móviles ha motivado el uso del procesamiento del lenguaje natural, especialmente, la tarea de identificación de partes del discurso (Part-of-Speech Tagging, POST), tanto como una tarea de preprocesamiento de las diferentes aplicaciones como del mismo etiquetado. Los enfoques tradicionales para la construcción de POST son costosos y dispendiosos, lo cual ha dado lugar a una nueva tendencia que busca propuestas que propendan por la sencillez y la eficiencia, como es el uso de algoritmos metaheurísticos, que han mostrado muy buen desempeño en comparación con los enfoques tradicionales.

Por tanto, en esta tesis se propone en primera instancia, un algoritmo memético, el cual es un tipo de algoritmo metaheurístico, que adicional al balance que hace entre búsqueda global y búsqueda local para encontrar soluciones, incluye conocimiento del problema. El algoritmo propuesto considera dos casos principales: Una lengua tradicional, como el inglés, y una no tradicional, como el nasa yuwe. En segunda instancia, buscando que el algoritmo propuesto para etiquetado pueda tener aplicaciones futuras en contextos locales, como es el caso de la revitalización de lenguas en peligro de exitinción, caso de la lengua nasa yuwe, la cual se beneficiará al obtener un POST, que puede ser utilizado en el desarrollo de objetos de aprendizaje complejos u otras utilidades. Por lo tanto, para complementar la propuesta se construyó el primer corpus etiquetado para nasa yuwe, que junto con el algoritmo memético de etiquetado propuesto se convierten en el primer acercamiento hacia aplicaciones de procesamiento de lenguaje natural sobre esta lengua. Sumado a lo anterior, se definieron una serie de experimentos para los dos casos, que muestran el desempeño del algoritmo propuesto en contraste con otros recomendados por la literatura.

El desarrollo de esta tesis estuvo enmarcado en la metodología Patrón de Investigación Iterativo, la cual permitió, en primera instancia, realizar un estudio exhaustivo sobre el estado del arte de las técnicas de construcción de POST, conjuntos de etiquetado y corpus utilizados, tanto para lenguas tradicionales como no tradicionales. En segunda instancia, a partir del estudio de los trabajos revisados y

centrando el interés de la investigación en POST, que utilizan algoritmos metaheurísticos se seleccionó el algoritmo metaheurístico Global Best Harmony Search como base para elaborar la propuesta del algoritmo de etiquetado. En tercera instancia, también a partir de la revisión de literatura realizada, fue posible obtener el corpus a utilizar para el idioma inglés, el conjunto de etiquetado universal, los aspectos relevantes a tener en cuentan en la construcción del corpus etiquetado de nasa yuwe, así como la selección y construcción de la línea base y la ejecución de los experimentos realizados sobre los corpus.

Como resultados concretos de esta tesis, en este documento se presenta en primer lugar, una breve reseña sobre el contexto y el estado del arte de los trabajos revisados, la cual se extiende en el Anexo A. En segundo lugar, se presentan los detalles de la construcción del corpus etiquetado para la lengua nasa, e información propia del corpus. En tercer lugar, se describe el algoritmo de etiquetado propuesto GBHS Tagger, con los detalles asociados con su construcción, las diferentes versiones por las que fue pasando hasta obtener la versión final, constituyéndose en el primer algoritmo memético utilizado para el problema de etiquetado. Finalmente, se presentan los diferentes experimentos realizados con el algoritmo propuesto en contraste con las líneas bases construídas, que permitieron comparar el desempeño del algoritmo propuesto sobre una lengua tradicional como el inglés, utilizando el corpus Brown, y una lengua no tradicional como es el nasa yuwe, para la cual no se contaba con ningún antecedente en esta área.

Los resultados obtenidos en el desarrollo de esta tesis, indican que el procesamiento de lenguaje natural es un área que tiene muchas aplicaciones actualmente, por tanto, el etiquetado se ha convertido en una fase importante para cualquiera de ellas, es así que tanto el corpus etiquetado para nasa yuwe, como el algoritmo memético de etiquetado propuesto, junto con los resultados de los experimentos, convierten a esta tesis en un referente en el problema de identificación de partes del discurso y aplicaciones, utilizando algoritmos meméticos. De tal forma, que la continuación de este trabajo se puede generar la construcción de corpus para lenguas que requieran revitalización, y visibilización a través de herramientas informáticas, u otras propuestas de algoritmos meméticos y metaheurísticos que sean robustas y eficientes como la propuesta en esta tesis.

Palabras Clave: Identificador de partes del discurso, etiquetador, algoritmo memético para etiquetado, conjunto de etiquetas, corpus etiquetado.

Structured abstract

The growth of communications through mobile devices has motivated the use of natural language processing. The traditional approaches to the construction of POST are costly and wasteful. This has led to a new trend, seeking proposals that favor simplicity and efficiency, such as the use of metaheuristic algorithms. These have performed well compared to traditional approaches.

This thesis thus first proposes a memetic algorithm, a type of metaheuristic algorithm, which in addition to the balance it creates between global and local search for finding solutions, includes knowledge of the problem. The proposed algorithm considers two main cases: a traditional language, such as English, and a non-traditional one, such as Nasa Yuwe. Secondly, the aim is that the tagging algorithm proposed will have applications in local contexts - such as is the case with revitalization of languages in danger of extinction. This is the situation with Nasa Yuwe, which will benefit from obtaining a POST that can be used in developing complex learning objects or other utilities. Therefore, to complement the proposal, the first annotated corpus for Nasa Yuwe was built, which together with the proposed tagging memetic algorithm becomes the first approach to natural language processing applications in this language. Furthermore, a series of experiments was defined for the two cases, which shows the performance of the proposed algorithm in contrast to others recommended by the literature.

This thesis was framed within the Iterative Research Pattern methodology. This allowed an initial exhaustive study on the state of the art of POST construction techniques, tagging sets and corpus used, for both traditional and non-traditional languages. Based on the work reviewed, with the focus on POST research using metaheuristic algorithms, the Global Best Harmony Search metaheuristic algorithm was then selected as the basis for carrying out the tagging algorithm proposal. Also based on the literature review, it was possible to obtain the corpus to be used for the English language, the universal tagging set, the relevant aspects to be taken into account in building the Nasa

Yuwe tagging corpus, as well as selection and construction of the baseline and execution of the experiments carried out on the corpus.

As concrete results of this thesis, this document presents a brief review of the context and the state of the art of the works reviewed, which is found in full in Annex A. The details of building the tagged corpus for Nasa Yuwe are then presented along with information specific to the corpus. The proposed tagging algorithm, GBHS Tagger, is then described with details relating to its construction and the different versions through which it passed to obtain the final version, constituting the first memetic algorithm used for the problem of tagging. Finally, the different experiments performed with the proposed algorithm are presented, contrasted with the baselines constructed, which made it possible to compare the performance of the proposed algorithm on a traditional language such as English, using the Brown corpus, and a non-traditional language such as Nasa Yuwe, for which there was no precedent in this area.

The results obtained in carrying out this thesis indicate that natural language processing is an area that has many applications today. As a result, tagging has become an important phase for any one of them. As such, both the tagged corpus for Nasa Yuwe and the proposed memetic tagging algorithm, together with the results of the experiments, make this thesis a reference in the problem of part-of-speech tagging and applications using memetic algorithms, so that the continuation of this work can generate the construction of corpus for languages that require revitalization, and visualization through computer tools or other proposals of memetic and metaheuristic algorithms that are as robust and efficient as that proposed in this thesis.

Keywords: Part-Of-Speech Tagging, tagger, memetic algorithm for tagging, tagset, annotated corpus, tagged corpus.

Contenido

Lista de figuras

Lista de tablas

Lista de algoritmos

Lista de anexos

Lista de ecuaciones

Listas de siglas

CRF: Condicional Random Fields – Campos Aleatorios Condicionales.

EFOs: Evaluaciones de la Función Objetivo.

GBHS: Global Best Harmony Search – Busqueda Global Armónica.

GBHS Tagger: Global Best Harmony Search Tagger – Algoritmo de etiquetado propuesto basado en GBHS.

HMM: Hidden Markov Models – Modelos ocultos de Markov.

HMCR: Harmony Memory Considering Rate -Tasa de consideración de la memoria armónica.

HMS: Harmony Memory Size – Tamaño de la memoria armónica.

HS: Harmony Search – Búsqueda armónica.

HS Tagger: Algoritmo de Etiquetado basado en Harmony Search.

MaxNeighbors: El número de vecinos que se evalúa en el proceso de optimización local del algoritmo GBHS Tagger.

MA: Memetic algorithm – Algoritmo memético.

MEMM: Maximum Entropy Markov Models, Modelo de Máxima Entropía.

NI: Number of Improvisations – Número de improvisaciones.

NLP: Natural Language Processing - Procesamiento de lenguaje natural

NLTK: Natural Language Toolkit.

PAR: Pitch Adjusting Rate - Tasa de Ajuste de Tono.

POST: Part-Of-Speech Tagging – Etiquetador - Identificador de Partes del Discurso.

ProbOpt: Probabilidad de optimización. Párametro que controla el porcentaje de veces que se realiza un proceso de optimización local en el algoritmo GBHS Tagger.

PSO: Particule Swarm Optimización.

S: representa la oración, que contiene las palabras a etiquetar.

$S = \{w_1, w_2, .., w_n\}$: secuencia de etiquetado, donde w_i indica la i-ésima palabra a ser etiquetada.

SVM: Support Vector Machines - Máquinas de soporte vectorial.

TnT: Trigram'sn'Tags – Trigramas.

T*: Representa la solución óptima, en relación a la secuencia correcta de etiquetas de una oración, con respecto a todas las otras posibles soluciones.

$U(1, N)$: Distribución uniforme entre 1 y N.

Glosario

Algoritmo memético: Es un algoritmo metaheurísto cuya población de agentes alternan períodos de auto-mejora (mediante la búsqueda local) con períodos de cooperación y competencia (búsqueda global: recombinación y selección) [1].

Algoritmo metaheurístico: Son algoritmos aproximados de optimización y búsqueda de propósito general. Son procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda [2].

Algoritmo metaheurístico de búsqueda aleatoria: La búsqueda aleatoria (Random Search, RS) [3] es un algoritmo de optimización estocástico y Global. Es un método de búsqueda directa, dado que evalúa soluciones (independientes entre sí) sobre todo el espacio de búsqueda usando una distribución de probabilidad uniforme. RS puede devolver una aproximación razonable a la solución óptima dentro de un tiempo adecuado para el caso de un problema de baja dimensionalidad, pero puede ser muy lento y pobre en calidad para problemas de alta dimensionalidad. En el **Algoritmo 1**, se presenta esta metaheurística.

Algoritmo 1. Algoritmo de Búsqueda aleatoria.

```
1. MejorSolución ←0;
2. Para cada iter_i ∈ NumIteraciones hacer
3.    candidata_i ← SolucionAleatoria(TamagnoProblema, EspacioBusqueda);
4.    Si Cost(candidata_i) < Cost(Mejor) entonces
5.          Mejor ← candidata_i;
6.    Fin si
7. Fin Para cada
8. devolver Mejor
```

Fuente: [3]

Algoritmo metaheurístico ascenso a la colina: Ascenso a la colina (Hill Climbing) [2]**,** es una técnica que no requiere conocimiento sobre la dirección de la solución, ya que evalúa iterativamente nuevas soluciones candidatas en el espacio de solución de la actual solución candidata y adopta la nueva solución cuando es mejor que la actual.

Esto favorece que escale la función hasta alcanzar un óptimo local. En el **Algoritmo 2**, se presenta esta metaheurística.

Algoritmo 2. Algoritmo Subiendo la Colina.

```
1. S ← algunaSoluciónInicial // procedimiento de inicialización
2. repetir
3.     R ← Tweak(Copia(S)) // procedimiento de modificación
4.     Si Calidad(R)> Calidad(S) entonces // procedimiento de evaluación y selección
5.         S ← R
6.     Fin Si
7. hasta que S sea la solución ideal o se acabe el tiempo
8. devolver S
```

Fuente: [2]

El procedimiento Tweak (modificar) suele ser una búsqueda estocástica (por ejemplo, un cambio aleatorio a la solución) de soluciones candidatas en el entorno de la solución actual que tengan mejor calidad.

Analizador léxico (tokenizers): Son programas que se encargan de hacer la descomposición léxica de un texto, es decir, la división de un texto en unidades léxicas conocidas como tokens [4]. Un analizador léxico tiene dos propiedades importantes: 1) Es el único componente que se debe usar en toda aplicación de procesamiento de lenguaje natural que involucre tareas de análisis. 2) Siempre se utiliza al inicio del proceso de análisis [5].

Búsqueda armónica (Harmony Search, HS): Algoritmo metaheurístico que se basa en el proceso de improvisación musical donde se busca obtener una armonía perfecta, es decir, un proceso de optimización que busca encontrar una solución global (un estado perfecto) determinado por una función objetivo [6].

Búsqueda armonica Global (Global-Best Harmony Search, GBHS): Algoritmo metaheurístico variante de HS, toma conceptos de optimización basada en partículas (Particle Swarm Optimization, PSO), para mejorar el desempeño de HS [7].

Colección de prueba: Es una colección de documentos, un conjunto de prueba de necesidades de información, expresadas como consultas de usuario y pueden estar constituidos por una o varias palabras clave y un conjunto de juicios de relevancia, evaluados generalmente de manera binaria, expresando si el documento es relevante o no para cada consulta. Las colecciones de prueba son la principal herramienta para hacer evaluación de los Sistemas de Recuperación de Información [8].

Colección de prueba de textos nasa yuwe: Esta conformada por 97 documentos de texto escritos en nasa yuwe, 8 consultas y los juicios expertos sobre las relevancias de estos documentos. Es la primera colección desarrollada para el nasa yuwe, construída para evaluar un sistema de recuperación de información para esta misma lengua [9].

Conjunto de etiquetas (Tagset): Es el conjunto de categorías o etiquetas posibles a asignar al conjunto de palabras de un texto [10]. El conjunto de etiquetas para cada lenguaje puede variar según los contextos y la estructura morfológica de cada lenguaje, es decir, se pueden apreciar variaciones y tendencias de unificación, así como diferentes métodos para hacer el etiquetado de las palabras que conforman los textos. Para el caso de esta tesis se utiliza el conjunto de etiquetado Universal propuesto por Petrov, que cuenta con 12 etiquetas posibles así: NOUN (*nouns*), VERB (*verbs*), ADJ (*adjectives*), ADV (adverbs), PRON (*pronouns*), DET (*determiners and articles*), ADP (*prepositions and postpositions*), NUM (*numerals*), CONJ (*conjunctions*), PRT (*particles*), '.' (*punctuation*) and X (*a catch-all for other categories such as abbreviations or foreign words*) [10].

Conjunto de datos (Dataset): Constituyen los datos propiamente dichos, sobre los cuales se va a realizar el procesamiento del algoritmo o modelo (según el caso). Se convierten en una parte muy importante de un sistema de aprendizaje automático (predicción, clasificación, entre otros). Estos datos son usados para el entrenamiento y las pruebas del sistema**.**

Corpus: Es una colección de textos auténticos legibles para una máquina (incluyendo transcripciones de datos hablados) los cuales son representativos de un lenguaje natural [11]**,** su contenido debe ser escogido para apoyar su propósito, como, por ejemplo, estudiar un lenguaje. En esta tesis se hablan sobre corpus etiquetados los cuales incluyen texto en donde cada palabra en una oración tiene su correpondiente etiqueta (tag o label) que señala el rol de la palabra en la oración (por ejemplo: nombre, verbo, entre otras). El rol proviene de un conjunto de etiquetas específico (tagset).

Espacio de búsqueda o espacio de soluciones: En un algoritmo metaheurístico, corresponde al dominio de la función que se desea optimizar, es decir, en donde el

algoritmo encuentra las posibles soluciones candidatas al problema, por lo tanto, es de vital importancia delimitar el espacio. Existen espacios de búsqueda discretos, binarios y continuos; para el problema de etiquetado tratado en esta tesis, el espacio de soluciones es discreto [12].

Etiquetador (Tagger): Algoritmo que permite hacer la Identificación de las partes del discurso de un texto.

Función de Calidad, Función Objetivo, Función Fitness o Función de Costo: En un algoritmo metaheurístco, es la medida cuantitativa del rendimiento de una solución que busca optimizar (maximizar o minimizar) uno o más objetivos. Algunos ejemplos de funciones objetivo son: la minimización de los materiales utilizados en la fabricación de un producto, la maximización del retorno de las inversiones en un portafolio, la maximización del beneficio de los objetos que se incluyen en un contenedor que se carga en un puerto marítimo, entre otros [12].

Identificación de partes del discurso (Part-Of-Speech Tagging, POST): Es el proceso de asignar etiquetas (tags) a cada palabra en un texto, habiendo realizado previamente un proceso de tokenización (análisis léxico). La entrada a un algoritmo etiquetador, es una secuencia de palabras y un conjunto de posibles etiquetas y la salida, es una secuencia de etiquetas asignadas en orden de acuerdo a la secuencia de palabras. La meta es encontrar la correcta etiqueta para cada palabra en diferentes contextos [13].

Matriz de confusión: Es una matriz que se asocia a los resultados entregados por un algoritmo de clasificación, que se usa para medir la precisión, el recuerdo (recall), la medida F, entre otras medidas de calidad y comparación de dichos resultados [14]. En dicha matriz, los verdaderos negativos, corresponde al total de instancias que siendo negativos han sido clasificados como negativas; los verdaderos positivos, corresponden al total de instancias que siendo positivas han sido clasificadas como positivas; los falsos positivos, corresponden al total de instancias que siendo negativas han sido clasificadas como positivas; y los falsos negativos, corresponden al total de instancias que siendo positivas han sido clasificadas como negativas. A continuación, en la **Figura 1** se puede observar un ejemplo de una matriz de confusión.

		Clase predecida	
		0	*1*
Clase Real	*0*	*Verdaderos Negativos (VN)*	*Falsos Positivos (FP)*
	1	*Falsos Negativos (FN)*	*Verdaderos Positivos (VP)*

Figura 1. Ejemplo de matriz de confusión. **Fuente**: [14]

Natural Language Toolkit (NLTK): Conjunto de módulos, corpus etiquetados, conjunto de etiquetado y tutoriales de lingüística computacional y procesamiento de lenguaje natural (PLN) para apoyo de la investigación y la enseñanza en PLN. Es un marco de trabajo totalmente escrito en Pyhton que provee varios algoritmos para la identificación de partes del discurso como el etiquetador Brill, el etiquetador basado en HMM, entre otros. Para el inglés, se encuentran los corpus Brown y Gutenberg, entre otros. NLTK cuenta con diversas implementaciones para trabajar con otras lenguas [15].

Palabras ambiguas (ambiguos words): Palabras que existen en una oración y según el contexto en el que se encuentre su etiqueta puede cambiar, es decir, son palabras que pueden tener más de una etiqueta [16]. Por ejemplo, la palabra *like,* puede tener etiquetas como verbo principal, preposición, conjunción, entre otras.

Patrón de Investigación Iterativa (Iterative Research Pattern): Esta metodología consta de 4 pasos básicos: observaciones de campo, identificación del problema, desarrollo tecnológico y pruebas de campo, soportado en los pasos del método científico y entendiendo por investigación (research) “al proceso de generar una pregunta y una hipótesis de investigación, diseñando y desarrollando un experimento y evaluando los resultados para responder la pregunta original y confirmar o negar la hipótesis” [17].

Precisión (Precision): Indica la proporción de elementos correctamente clasificados que son verdaderos positivos (VP), ver **Ecuación 1**.

$$Precisión = \frac{VP}{(VP + FP)}$$

Ecuación 1. Fórmula de Precisión**. Fuente**: [18]

Donde, VP (Verdaderos Positivos), es el número de instancias que son correctamente clasificados, FP (Falsos Positivos) es el número de instancias cubiertos por la regla y

que son incorrectamente clasificados. Para el caso, de esta tesis, la precision se calcula conforme a la **Ecuación 2**.

$$Precision\ = \frac{(\#\ palabras\ correctamente\ etiquetadas)}{(\#\ palabras)} * 100$$

Ecuación 2. Fórmula de Precisión. **Fuente*:*** [19]

Recuerdo (Recall, Sensitivity): Es la porción de instancias positivas que han sido correctamente clasificadas sobre el total de las instancias. Se calcula según la **Ecuación 3**.

$$Recuerdo = \frac{VP}{(VP + FN)}$$

Ecuación 3. Fórmula de Recuerdo. Fuente: [14]

En donde, VP (Verdaderos positivos), corresponde al total de instancias que siendo positivas han sido clasificadas como positivas y FN (Falsos negativos), corresponde al total de instancias que siendo positivas han sido clasificadas como negativas.

Teorema del límite central: Un conjunto de variables independientes con la misma distribución, el promedio y la varianza se comporta cocmo una distribución normal [20].

Validación cruzada usando K grupos (K-fold cross-validation): Es un proceso de validación de modelos comúnmente usado en minería de datos y áreas relacionadas, en la que los datos se dividen en K subconjuntos (folders). Uno de los subconjuntos se utiliza como datos de prueba y el resto $(K - 1)$ como datos de entrenamiento. El proceso de validación cruzada es repetido durante K iteraciones, con cada uno de los posibles subconjuntos de datos como dataset de prueba [21].

Validación cruzada dejando uno fuera (Leave-One-Out Cross-Validation): Es un proceso de validación de modelos similar a la validación cruzada usando K grupos, pero el K grupos corresponde a los N registros $(K = N)$. esto implica que los datos se separan de forma tal que en cada iteración se tiene un solo dato de prueba y el resto de datos conforman el dataset de entrenamiento. Tiene dos ventajas: 1) La estimación del error es mucho más estable; y 2) No se sobreestima el error. Tiene una desventaja importante: el tiempo de ejecución puede ser muy alto [22].

Capítulo 1

Introducción

1.1 Presentación

Como temas centrales de esta tesis, se tiene en primera instancia, el problema de identificación de partes del discurso (Part-of-Speech Tagging – POST) y en segunda instancia, la propuesta de un algoritmo memético para resolver el problema de etiquetado, lo cual fue motivado dada la complejidad del problema de etiquetado, su aplicación en diferentes tareas de procesamiento de lenguaje natural y la tendencia en buscar soluciones para este problema, menos complejas, pero con buenos resultados. Durante la revisión de la literatura sobre los diferentes enfoques de construcción de POST, se pudo apreciar que, el uso de algoritmos metaheurísticos en este problema ha obtenido resultados competitivos, comparados con los enfoques tradicionales. Lo anterior, sumado al hecho, de que la inclusión de conocimiento y optimización local algoritmos metaheurísticos en otros problemas de optimización (algoritmos meméticos), reportó resultados sobresalientes, lo cual dio lugar a proponer la construcción de un POST basado en algoritmos meméticos.

Para hacer la evaluación del POST construido se seleccionaron dos casos así:

- En primera instancia, se utilizó un caso de una lengua tradicional, el inglés, con el objetivo de comparar y contrastar los resultados obtenidos en este proyecto frente al estado del arte, así como definir opciones de trabajo futuro con otras lenguas tradicionales.

- En segunda instancia, una lengua no tradicional fue seleccionada, como es el caso del nasa yuwe, dado que es una lengua que no cuenta con muchos recursos lingüísticos, debido a su condición de lengua independiente y que actualmente, continúa en proceso de descripción. El nasa yuwe, es una de las lenguas oficiales de la República de Colombia [23], que por diversos factores históricos la han llevado actualmente ha ser clasificada como una lengua en peligro de extinción. Por tanto, se han venido definiendo diferentes estrategias para su revitalización,

entre las que se cuentan el uso de recursos informáticos para su enseñanza y la generación de recursos lingüísticos que ofrezcan condiciones para que el nasa yuwe continúe adelantando procesos de visualización y sensibilización a través de estrategias computacionales aplicables en diferentes áreas.

1.2 Planteamiento del problema

El lenguaje natural juega un papel muy importante en la comunicación entre los seres humanos. En los últimos años, se ha observado un crecimiento acelerado en las tecnologías de la información y las ciencias de la computación, las cuales han cambiado significativamente varios aspectos de las actividades de la vida de las personas, como es el caso de las comunicaciones verbales y escritas a través de aplicaciones que usan el computador.

Ante este panorama, el procesamiento de lenguaje natural día a día se ha convertido en un tópico de investigación cada vez más importante, especialmente los Identificadores de Partes del Discurso (Part of Speech Tagging, POS Tagging) [24], dado que son un paso a considerar en el pre-procesamiento que se realiza al texto en sistemas de Lenguaje Natural, proveyendo información útil sobre el vocabulario y su contexto reduciendo o eliminando ambigüedades que son clave para diferentes tipos de aplicaciones como: sistemas de reconocimiento de voz, conversión de texto a voz, traducción automática, clasificación de textos, extracción automática de información, recuperación de información multimedia, análisis de sentimiento, resolución de ambigüedades en el significado de las palabras en un contexto, entre otras [25].

Desarrollar un Identificador de partes del discurso no es un proceso fácil debido a que en todos los lenguajes existe ambigüedad en el uso de las palabras [19], por lo tanto, la identificación de la etiqueta y la cantidad de etiquetas que se le pueden asociar a una palabra se considera un problema complejo, sin embargo, como se mencionó anteriormente, es necesario contar con identificadores de partes del discurso más precisos, dada la diversidad de sus usos y aplicaciones.

En general, los etiquetadores pueden ser vistos como un problema de aprendizaje supervisado [26], aplicando varios modelos como: los basados en el uso de información estadística (Modelos ocultos de Markov [27], Trigramas [28], Máxima Entropía [29, 30], entre otras), el aprendizaje basado en transformación de reglas [31],

redes neuronales [32], árboles de decisión [33], etc., los cuales han avanzado en el estado del arte mostrando resultados con alta precisión (especialmente los que utilizan información estadística). Sin embargo, los métodos supervisados dependen del etiquetado de los datos de entrenamiento, haciendo que el proceso sea complejo y computacionalmente costoso.

En recientes trabajos, se reporta el uso de enfoques no supervisados y semi-supervisados, como una alternativa para solucionar el problema de etiquetado manual de los datos de entrenamiento, como por ejemplo, la proyección de etiquetas basadas en grafos [34], redes neuronales recurrentes [35], en los cuales se presentan problemas de demasiado ruido en los datos, complejidad, alta dependencia de los datos, ajuste complejo de parámetros para el entrenamiento y etiquetado, entre otros, haciendo que en algunos casos los valores de precisión sean poco competitivos.

También se han utilizado algoritmos metaheurísticos para realizar la búsqueda de la etiqueta que con una mayor probabilidad se debe asignar a una palabra según su contexto, obteniendo algoritmos eficientes que alcanzan resultados muy competitivos en tiempos de cómputo razonables [19].

Teniendo en cuenta que: i) el debate sobre cuál enfoque de solución es el mejor para el problema de etiquetado no ha finalizado, ii) en recientes comparaciones se empieza a dar más valor a aquellos enfoques que son más sencillos [36, 37] (basado en racero de ockham [38]: “cuando dos o más explicaciones se ofrecen para un fenómeno, la explicación completa más simple es preferible; es decir, no deben multiplicarse las entidades sin necesidad”), iii) el uso de algoritmos metaheurísticos en la solución de problemas altamente complejos de la vida real [3, 39] entre los que se incluyen el procesamiento de lenguaje natural y el etiquetado de palabras [19, 26, 40, 41, 42, 43], ha mostrado resultados competitivos comparados con otras aproximaciones y son más sencillos de diseñar e implementar, por esto, en el presente trabajo se considera apropiado abordar el problema de etiquetado desde este enfoque de solución, entendiendo que al día de hoy no se ha definido cual metaheurística es la mejor para resolver este tipo de problemas.

Además, considerando el segundo teorema de No Free Lunch Theorems for Optimization (NFLT) [44] que establece que “un algoritmo metaheurístico puede superar a otro en un problema cuando ninguno de ellos está especializado en el

problema"[1], obliga a considerar que la inclusión de conocimiento específico del problema debe ser considerado para la obtención de mejores resultados en un problema de optimización. En este sentido, los algoritmos meméticos, una clase especializada de metaheurística, que combina búsqueda global, búsqueda local y conocimiento del problema [45, 46, 47, 48] deben ser considerados en la identificación de partes del discurso, ya que a la fecha no se reportan en la solución de este problema y los antecedentes en otras áreas son muy promisorios.

Basado en lo anteriormente expuesto, se da origen a la siguiente pregunta de investigación: ¿Cómo mejorar la tarea de etiquetado de partes del discurso mediante un algoritmo que haga un mejor balance entre la exploración y la explotación soportado en conocimiento específico del problema?

En busca de dar solución a la anterior pregunta, es posible pensar en proponer un algoritmo más eficiente para encontrar la etiqueta más probable para una palabra dentro de una oración específica (contexto) [39] utilizando un algoritmo memético. Un algoritmo que haga un mejor balance entre la exploración y la explotación soportado en conocimiento específico del problema [44] que permita obtener mejor precisión en los resultados del etiquetado en un menor tiempo de ejecución.

1.3 Objetivos

A continuación, se presentan los objetivos como fueron aprobados por el Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca en el documento de la propuesta de tesis, que buscan resolver lo presentado en la sección anterior, planteamiento del problema.

1.3.1 Objetivo general

Proponer un algoritmo basado en meméticos para encontrar la etiqueta más probable para una palabra dentro de una oración específica (contexto) para los casos de una lengua tradicional y una no tradicional.

1.3.2 Objetivos específicos

- Proponer un algoritmo basado en meméticos que permita realizar la tarea de identificación de partes del discurso buscando priorizar la sencillez de la propuesta.

[1] Interpretación del Teorema [42]

- Diseñar y Construir un Corpus de etiquetado manual para la lengua nasa yuwe[2] teniendo en cuenta la descripción presentada por [49] y el etiquetado Universal [10], que pueda ser utilizado en la tarea de Identificar partes del Discurso.

- Evaluar el nivel de desempeño del identificador de partes del discurso construido considerando la precisión y recuerdo de las etiquetas asignadas, mediante dos casos de estudio: 1) Una lengua tradicional, como el inglés, utilizando un corpus lingüístico ampliamente utilizado para la tarea de etiquetado, y 2) Una lengua no tradicional, como el nasa yuwe, utilizando el corpus lingüístico construido.

1.4 Contribuciones de esta tesis

En esta sección, se presentan las contribuciones que el desarrollo de esta tesis hace a las áreas de procesamiento del lenguaje natural y algoritmos meméticos.

- El aporte central, es la propuesta de un algoritmo de etiquetado basado en meméticos, el cual mostró buen desempeño con el corpus Brown para el idioma inglés, en comparación con los algoritmos de la línea base establecida. Esta propuesta genera nuevo conocimiento, útil para la comunidad académica y científica nacional e internacional como base de nuevas investigaciones en procesamiento de lenguaje natural y la construcción de POST basado en algoritmos. Los resultados de esta propuesta fueron publicados como "**Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging**" en: Mining Intelligence and Knowledge Exploration - MIKE 2017. Con memorias en la revista Lecture Notes in Computer Science, vol 10682, Springer que para el 2017 esta indexada en categoría **A2** por el PUBLINDEX de Colciencias. Este artículo fue invitado a presentar una versión extendida para la revista IDRBT Journal of Banking Technology. Es pertinente resaltar que como parte del trabajo introductorio para el desarrollo de esta tesis se obtuvo un artículo titulado "**Continuous Optimization Based on a Hybridization of Differential Evolution with K-means**", publicado en 2014 en la conferencia IBERAMIA como Lecture Notes in Computer Science vol 8864, Springer, clasificado como A2 por Colciencias, el cual permitió conocer y aprender sobre los algoritmos

[2] Lengua de la comunidad indígena Páez (nasa), de tradición oral [126], es una lengua pobre en recursos lingüísticos.

metaheurísticos, sus fortalezas, aplicaciones y proponer una mejora a una estrategia de Evolución Diferencial.

- Como segundo aporte, esta tesis presenta la construcción de un corpus lingüístico etiquetado para nasa yuwe, el cual no existía, y se convierte en la base para continuar trabajando en diferentes aplicaciones de procesamiento del lenguaje natural en esta lengua. Los textos que conforman este corpus etiquetado fueron tomados de la colección de prueba para nasa yuwe, como resultado de un trabajo previo realizado con la lengua nasa, el cual fue publicado como "**Building a nasa yuwe language test collection**" en el 16th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2015. Evento con memorias en la revista Computación y Sistemas, que en 2015 tambien fue categorizada como **A2** por Colciencias.

- El tercer aporte, consiste en la aplicación del algoritmo memético propuesto en el corpus etiquetado para nasa yuwe, dado que es la primera vez que se realiza este tipo de procesamiento sobre esta lengua. Se considera que los resultados obtenidos, pueden establecer una línea base para el desarrollo de futuras propuestas tanto de etiquetadores como de mejoras al corpus, lo cual, favorecerá el desarrollo de aplicaciones más elaboradas que impulsen la revitalización de esta lengua. También proveerán una guía para futuros trabajos sobre otras lenguas que se encuentren en la misma situación del nasa yuwe.

- El proceso de construcción del corpus etiquetado para nasa yuwe y los experimentos realizados sobre este corpus con los etiquetadores de la línea base y el etiquetador basado en algoritmos meméticos propuesto en esta tesis, se han plasmado en un artículo titulado "**Building a Nasa Yuwe Language Corpus and tagging with a metaheuristic approach**", el cual fue aceptado en la 19th International Conference on Computational Linguistics and Intelligent Text Processing. Conferencia que cuenta con publicaciones indexadas A2 por Colciencias. Este artículo fue aceptado, se encuentra pendiente de publicación; se puede apreciar en el Anexo D, junto con la aceptación. Adicionalmente, como parte de un trabajo previo con la lengua nasa yuwe, durante el desarrollo del doctorado, se obtuvo un artículo titulado "**Tokenizer adapted for nasa yuwe language**", publicado en 2016 en la Revista Computación y Sistemas, clasificada como A2 por

Colciencias, este artículo es un gran aporte en el trabajo futuro de esta tesis dado que presenta un tokenizador que permite dividir las oraciones escritas en nasa en los correspondientes tokens, lo cual al momento de integrar el etiquetador propuesto en esta tesis, con una herramienta software, apoya la tarea de etiquetado, separando correctamente las palabras que conforman una oración escrita en nasa yuwe, antes de asignarles su correspondiente etiqueta.

- Finalmente, se elaboró una detallada revisión del estado del arte sobre técnicas de construcción de etiquetadores, haciendo especial énfasis en el uso de algoritmos metaheurísticos. Este trabajo permite obtener una visión global y detallada sobre sobre el estado actual de la temática, favoreciendo el planteamiento y desarrollo de futuras investigaciones. Este estado del arte se encuetra en revisiones finales para su posterior envío a una revista especializada. El artículo se titula "**A review on Building Part-Of-Speech Tagging**" y se puede observar en el Anexo A del presente documento.

Adicionalmente a las contribuciones antes mencionadas, el desarrollo de esta tesis, permitió fortalecer las líneas de investigación de los grupos de investigación: Grupo de Ingeniería Telemática (GIT), Grupo de I+D en Tecnologías de la Información (GTI) y Grupo de Estudios Lingüísticos, Pedagógicos y Socioculturales del Suroccidente colombiano (GELPS) de la Universidad del Cauca, así como el trabajo interdisciplinario que favorece el aporte a la revitalización de lenguas y la construcción de conocimiento desde diferentes perspectivas.

1.5 Organización del documento

En esta sección se presentaron el planteamiento del problema, los objetivos y las contribuciones a nivel de investigación que se obtuvieron con el desarrollo de esta tesis doctoral.

Seguidamente, en el capítulo 2, se presenta un marco de referencia, el cual incluye un contexto teórico y los antecedentes más relevantes relacionados con los temas tratados en este trabajo.

Luego, el capítulo 3, está dedicado a mostrar los corpus de etiquetado de las lenguas utilizadas en esta tesis, de tal forma que, en primera instancia, se presenta el proceso de construcción del corpus etiquetado para nasa yuwe, junto con los resultados

obtenidos y la alineación del corpus al etiquetado universal. En segunda instancia, se presenta una breve descripción del corpus de etiquetado Brown, para la lengua inglés.

En el capítulo 4, se introduce el algoritmo Memético propuesto para el problema de etiquetado, junto con las líneas bases construidas.

En el capítulo 5, se describen los experimentos realizados, los resultados obtenidos y las discusiones sobre los mismos.

Finalmente, en el capítulo 6, se presentan las conclusiones alcanzadas con el desarrollo de este trabajo y las propuestas para el desarrollo de futuras investigaciones.

Capítulo 2

Estado del arte

En este capitulo se ha realizado una recopilación de los conceptos abordados como tema central de este documento. En la sección 2.1, se presenta un contexto que permite familiarizarse con el tema central abordado; en la sección 2.2, se hace referencia de manera muy resumida a trabajos encontrados en la literatura relacionados con la investigación y que aportan conocimiento sobre el estado actual de la temática y las brechas existentes. En la sección 2.3, a modo de resumen, se hace un análisis comparativo sobre los trabajos presentados en la sección 2.2. Finalmente, se presenta una breve síntesis sobre los trabajos revisados.

2.1 Contexto

En esta sección, se presentan algunos conceptos relevantes en el desarrollo de la presente investigación, como son: construcción de POST, algoritmos metaheurísticos, y una breve descripción de la lengua nasa. A continuación, se describen varios enfoques utilizados para la construcción de etiquetadores.

2.1.1 Identificadores de partes del discurso

Los Identificadores de Partes del Discurso (Part-of-Speech Tagging, POST) o Etiquetadores (Taggers) son una herramienta muy utilizada en el procesamiento de lenguaje natural, dado que su uso ha tomado un rol determinante al proveer información útil sobre el vocabulario y su contexto, en una gran variedad de aplicaciones [25].

Identificar las partes del discurso es el proceso de marcar palabras en un texto (corpus) basándose en su relación con las otras palabras que se encuentran alrededor (en su contexto) [25]. A cada palabra se le asigna su correspondiente categoría morfosintáctica (por ejemplo: nombre, verbo, adjetivo, etc.). En el problema de etiquetado existen basicamente algunas dificultades [16]: 1) Palabras ambigüas, en una oración existen algunas palabras que tienen más de una etiqueta posible. 2) Palabras desconocidas, son palabras que no se encuentran entre las reglas de los POST basados en reglas o entre el corpus de entrenamiento, lo cual se convierte en un punto critico en el diseño de un etiquetador. 3) Otro problema del etiquetado, que

no se relaciona con el lenguaje, es la consistencia del conjunto de etiquetas (tagset), dado que usar un conjunto grande de etiquetas, permite conocer más sobre las estructuras morfológicas y morfosintácticas de las palabras, pero dificulta la tarea de distinguir entre etiquetas similares.

Los sistemas que automáticamente asignan partes del discurso a las palabras en un texto deben tener en cuenta tanto el léxico como las limitaciones contextuales [50] y se basan generalmente en la propuesta de [31] que consta de dos estados: Uno que hace la clasificación de las palabras desconocidas usando información morfológica y el otro que clasifica las palabras conocidas y desconocidas usando información contextual (desambiguación) [51]. El conjunto de categorías de palabras (tagset) se fija en unas categorías específicas de etiquetado, sin embargo, cada conjunto de datos (dataset) provee diferentes etiquetas, variando la granularidad del Identificador de Partes del Discurso, es decir, algunos incluyen una categoría general para el etiquetado de verbos, mientras otros incluyen varias categorías específicas como: verbo principal, verbo auxiliar, entre otros [52].

Desarrollar un Identificador de partes del discurso no es un proceso fácil [19], como se mencionó anteriormente, sin embargo, contar con identificadores de partes del discurso más precisos es muy importante, dada la diversidad de sus aplicaciones. En general, para la construcción de los etiquetadores se utilizan varios enfoques, entre ellos, los más importantes son los etiquetadores basados en reglas y los etiquetadores basados en métodos estadísticos.

A continuación, se mencionan algunos enfoques de construcción de etiquetadores de partes del discurso:

Etiquetadores basados en reglas. Los etiquetadores basados en reglas definen la forma en que a cada palabra se le asigna su correspondiente etiqueta empleando reglas que se aplican a una secuencia dada de palabra de entrada [19]. La asignación inicial se realiza mediante estadísticas simples y posteriormente en cada iteración de aprendizaje, se van seleccionando las reglas con más alto puntaje que van siendo aplicadas a los datos de entrenamiento, esto permite generar una lista ordenada de reglas y, por tanto, se van actualizando los datos de entrenamiento usando la regla seleccionada [33]. Estos métodos involucran gran trabajo debido a que la escritura de reglas implica conocimiento profundo del lenguaje y, además, computacionalmente esta tarea generalmente consume demasiado tiempo [33].

Etiquetadores basados en métodos estadísticos. Estos etiquetadores han sido una aproximación fundamental en lingüística computacional, proporcionando avances significativos en la solución del etiquetado de palabras, entre los métodos más usados, se encuentran:

- Los Modelos ocultos de Markov (Hidden Markov Model, HMM), que son el enfoque más usado, el cual requiere de información contextual de la lengua, pero poco conocimiento sobre esta [19]; estos etiquetadores utilizan los conceptos de HMM para encontrar la secuencia de etiqueta más probable para cada oración, que maximiza la siguiente fórmula: $P(w|t) * P(t|t_{-1}, t_{-2}, \dots, t_n)$ [53], estas probabilidades pueden ser calculadas directamente sobre un corpus previamente etiquetado.

- Trigramas (Trigram'sn'Tags, TnT) [28], que utilizan Modelos de Markov de segundo orden para el diseño del POST, buscando dos palabras atrás. Los estados del modelo representan las etiquetas, las salidas representan las palabras, las probabilidades de transición depende de los estados (pares de etiquetas), las probabilidades de salida solo dependen de las categorías más recientes. Las probabilidades de salida y transición son estimadas desde un corpus previamente etiquetado.

- Modelo de Máxima Entropía (Maximum Entropy Markov Models, MEMM) [30] que utiliza una secuencia de modelos probabilísticos condicionales, en donde cada estado de salida tiene un modelo exponencial que toma las características observadas como entradas, y las salidas corresponden a una distribución sobre posibles próximos estados. Estos modelos exponenciales se entrenan por un método apropiado de escalamiento iterativo en el framework de máxima entropía.

- Campos Aleatorios Condicionales (Conditional Random Fields, CRF) [30], que corresponden a un modelo de estado finito con probabilidades de transición no normalizadas, asignando una distribución de probabilidad bien definida sobre posibles etiquetas, entrenadas por probabilidad máxima. CRF aplica un solo modelo exponencial para la probabilidad conjunta de la secuencia entera de etiquetas dada la secuencia de observación.

Otros enfoques de construcción de etiquetadores. En la literatura también se reportan otros enfoques como:

- Redes neuronales [33], los cuales generalmente utilizan un algoritmo de propagación hacia atrás (backpropagation) para el entrenamiento y alcanza muy buenos resultados si se tienen grandes cantidades de datos de entrenamiento con buen nivel de calidad.

- Árboles de decisión (Decision Trees, DT) [16], son utilizados para tareas de clasificación y de manera similar a los etiquetadores basados en reglas, pueden tener en cuenta el contexto y facilitar representaciones de características.

- Propagación de etiquetas basada en Grafos (Graph-based label propagation) [34], se utiliza en general para construir un etiquetador de palabras entre varios lenguajes, que puede ser basado en métodos tradicionales (TnT, HMM) u otros métodos como redes neuronales recurrentes, en combinación con un método de proyección de etiquetas (un algoritmo, o grafo), etiqueta las palabras de otro lenguaje a partir de la alineación de corpus.

- Máquinas de soporte vectorial (Support Vector Machines, SVM), es un clasificador originalmente diseñado para problemas de dos clases que maximizan la distancia entre puntos cercanos de las dos clases [54]. Para el problema de etiquetado SVM tiene ventajas como son: pueden manejar fácilmente gran cantidad de características (espacios con muchas dimensiones) y son resistentes al sobreentrenamiento (overfiting) [55].

- Algoritmos Metaheurísticos [26], que utilizan información estadística o basada en reglas, en el primer caso el algoritmo evoluciona para asignar la etiqueta más probable de una palabra en una oración basada en la tabla de contexto que tiene la misma información que en las técnicas estadísticas; en el segundo, el algoritmo metaheurístico se utiliza para evolucionar reglas de manera similar a como lo propone Brill [31].

2.1.2 Algoritmos Metaheurísticos

Los algoritmos metaheurísticos se utilizan para la solución de problemas de optimización o búsqueda empleando métodos computacionales, en general, combinan la búsqueda aleatoria, dada por las transformaciones de la población, con una búsqueda dirigida dada por la selección [3]. Dichas operaciones permiten explorar el espacio de búsqueda evitando el estancamiento en una solución óptima local y

acceder rápidamente a diferentes regiones del espacio de búsqueda de manera eficiente.

Entre los algoritmos metaheurísticos, en primera instancia, se encuentran los algoritmos de estado simple [2] como: Ascenso a la colina (Hill Climbing), Recocido Simulado (Simulated Annealing), Búsqueda Tabú (Tabu Search), Búsqueda local iterativa, Búsqueda local de vecindad variable (Variable Neighborhood Search), Búsqueda aleatoria (Random Search), entre otros. En segunda instancia, se encuentran los métodos basados en población [2] [3] como: Algoritmos Genéticos (Genetic Algorithm), Evolución Diferencial (Differential Evolution), Optimización basada en inteligencia de enjambres (Particle Swarm Optimization - PSO), Búsqueda Armónica (Harmony Search - HS), Búsqueda Cucú (Cucu Search), Búsqueda basada en colonias de hormigas (Ant Colony), Algoritmos meméticos (Memetic algorithms), entre otros.

Estos últimos, utilizan una población de individuos, que representan soluciones candidatas a un problema, la cual se somete a ciertas transformaciones y después a un proceso de selección, que favorece a los mejores. Cada ciclo de transformación y selección constituye una generación, de forma que después de cierto número de generaciones se espera que el mejor individuo de la población esté cerca de la solución buscada.

A continuación, se describen algunos algoritmos metaheurísticos utilizados en el desarrollo de este trabajo.

Búsqueda armónica (Harmony Search, HM): Es una metaheurística evolutiva que imita el proceso de improvisación que utilizan los músicos de jazz para buscar las mejoras armonías en sus canciones [56]. Fue propuesto en 2005 por Lee and Gee [6]. HS se orienta por los parámetros: Tasa de consideración de la memoria armónica (harmony memory considering rate, HMCR) y tasa de ajuste de tono (pitch adjusting rate, PAR) que controlan la búsqueda local y global, una distancia de ancho de banda arbitraria (bandwidth, bw) para cambiar o mutar algunos valores en las nuevas soluciones (armonías), tamaño de memoria armonica (harmony memory size, HMS), que define el tamaño de la memoria armónica (población) y el número máximo de improvisaciones (Number of improvisations, NI) que se usa como criterio de parada. El algoritmo consta de los siguientes pasos [6]:

1) Inicialización del problema de optimización y los parámetros del algoritmo, incluyendo si es de maximización o minimización, la función de aptitud (función

fitness o función objetivo), el rango de las variables, el tamaño de la memoria armónica, entre otros.

2) Inicialización de la memoria armónica (harmony memory, HM), que normalmente se hace aleatoriamente.

3) Generación de una nueva armonía o improviso (paso de improvisación) teniendo en cuenta tres reglas: Consideración de la memoria, Ajuste de tono y selección aleatoria.

4) Actualización de la HM: el nuevo vector (improviso) generado reemplaza la peor armonía en la HM si su fitness es mejor que el del peor. 5) Se repiten los pasos 3 y 4 hasta completar el criterio de terminación o cuando se alcanzan el número de improvisaciones (NI) definidas en el paso 1.

En 2007, Mahdavi et.al, [7] propusieron una mejora denominada improved harmony search algorithm (IHS), el cual plantea dos ecuaciones para actualizar dinámicamente el parámetro PAR y el bw, los cuales tiene un importante efecto sobre el desempeño de HS.

En 2008, Omran et.al, propusieron Global-Best Harmony Search (GBHS) algorithm [57], el cual hibrida la búsqueda armónica HS con el concepto de inteligencia de enjambre propuesto en PSO [58], donde un enjambre de individuos (llamados partículas) vuela a través del espacio de búsqueda. GBHS modifica el paso de ajuste de tono en HS de modo que la nueva armonía puede imitar la mejor armonía en la memoria armónica. Esto permite a GBHS trabajar más eficientemente en problemas continuos, binarios y discretos que HS e IHS [57], por este motivo fue seleccionado como la base para la presente propuesta. En el Algoritmo 2.1, se observa un resumen del algoritmo GBHS.

Algoritmo 2.1 Global Best Harmony Search.

1. Inicializacion del problema (Maximización o Minimización) y los parámetros HS: HMS, HMCR, PAR y NI
2. Inicializar HM
3. **repetir** /* improvisar una nueva armonía */
4. **para cada** $i \in [1, \mathrm{N}]$ **hacer**
5. **Si** $U(0,1) < HMCR$ **entonces** /* *consideración de la memoria* */
6. **Inicio**
7. $x_i' = x_i^j$, donde $j \sim U(1, HMS)$
   Si $U(0,1) \leq PAR(t)$ **entonces** /* *tono de ajuste para la generación (IHS)*/
8. $x_i' = x_k^{Mejor}$, donde $Mejor$ es el índice de la Mejor armonía en HM y $k \sim U(1, N)$

Algoritmo 2.1 Global Best Harmony Search.

```
9.          Fin
10.         Sino /*Selección aleatoria*/
11.             x_i' = LB_i + r × (UB_i − LB_i)
12.         Fin Si
13.     Fin para
14.     Actualizar HM /* Se reemplaza la peor armonía */
15. hasta que el NI es alcanzado
```

Fuente: [57]

Memoria Tabú: La Búsqueda Tabú (Tabú Search, TS) fue propuesto por Fred Glover en 1997 [59] y ha obtenido éxito en la solución de problemas de optimización duros (NP-hard). Este algoritmo utiliza una memoria adaptativa que tiene en cuenta soluciones que se han explorado anteriormente, durante el proceso de resolución del problema, por tanto, favorece la implementación de procedimientos para hacer explotación o no de el espacio de soluciones que se encuentra cerca de las soluciones que se van insertando en la memoria adaptativa, de tal forma, que la búsqueda de soluciones se convierte en un proceso eficaz y eficiente [60].

La memoria de la búsqueda tabú funciona mediante cuatro dimensiones principales, como son: la propiedad de ser reciente, la frecuencia, la calidad y la influencia. Las memorias basadas en lo reciente y en la frecuencia se complementan para obtener balance entre intensificación y diversificación. La dimensión de calidad hace referencia a la habilidad para diferenciar la bondad de las soluciones visitadas. La dimensión de influencia considera el impacto de las decisiones tomadas durante la búsqueda, tanto en calidad como en su estructura [60].

El uso de la memoria tabú puede ser explícita o implícita, es decir, la memoria explícita almacena soluciones completas, que puede ser utilizado para evitar visitar las mismas soluciones una y otra vez, también permite expandir los espacios de búsqueda durante la búsqueda local. Mientras que la memoria implícita (o basada en atributos) almacena información sobre determinados atributos de las soluciones, los cuales pueden cambiar de una solución a otra, también reduce los espacios de búsqueda evitando ciertos movimientos en la búsqueda local [60].

Algoritmos Meméticos. Existen varias formas de combinar un algoritmo de explotación (búsqueda local) con un algoritmo exploratorio (búsqueda global). Por ejemplo, Hill Climbing con reinicios [2], que combina un algoritmo de búsqueda local (Hill-Climbing) con un algoritmo global (Búsqueda aleatoria). De hecho, el algoritmo de mejora local ni siquiera tiene que ser metaheurístico, puede ser un algoritmo de

aprendizaje automático o un algoritmo heurístico. Este tipo de combinaciones son llamados Algoritmos Meméticos (Memetic Algorithms, MA) [3]. Los MA combinan estos dos métodos de búsqueda (global y local) para tomar las ventajas de estas dos estrategias. La búsqueda basada en población que permite la exploración de soluciones y la búsqueda local basada en vecindad que permite la explotación sobre soluciones prometedoras [61, 62]. El primer algoritmo MA implementado fue en el contexto del problema del Vendedor Viajero (Travelling Salesman Problem, TSP) [48].

De manera general, un MA se compone de [63]:

1) Un conjunto de soluciones candidatas (población de individuos) para probar el espacio de búsqueda.
2) Un operador de combinación (crossover) para crear nuevas soluciones candidatas (descendencia) mediante la combinación de dos o más soluciones existentes.
3) Un operador para mejorar las soluciones (descendencia).
4) Una estrategia de administración de la población.
5) Una función de evaluación (o función fitness) para evaluar la calidad de cada solución candidata, un mecanismo de selección para determinar cuales soluciones candidatas sobrevivirán y se someterán a variaciones.

Desde una perspectiva operacional, un MA típico, inicia con una población y luego repite los ciclos de evolución. Cada ciclo (también llamado generación) consiste de 5 pasos secuenciales [63]:

1) Selección de padres, que determina las soluciones candidatas que serán usadas para crear las nuevas soluciones, basado en el valor de la función de aptitud o un criterio de diversidad. Algunas de las estrategias de selección más comunes son: ruleta, torneo y elitismo. La selección también puede realizarse de acuerdo a un criterio de diversidad, en este caso se permiten individuos distantes para la reproducción.
2) Cruce de padres para generar la descendencia, que crea nuevas soluciones candidatas prometedoras, dirigiendo el proceso de optimización a nuevas áreas de búsqueda que permiten encontrar mejores soluciones, para lograr esto es necesario capturar conocimiento del problema.

3) Mutación, el operador de mutación puede ser aplicado para reforzar la diversidad de la población, aunque no es necesario debido a que la búsqueda local puede ser vista como un operador de macro mutación guiada.

4) Mejora local, mejora la calidad de la descendencia iterativamente reemplazando la solución actual por una solución tomada de la vecindad, este proceso se detiene para dar la mejor solución cuando se cumple con una condición de parada. La búsqueda local juega el rol de intensidad de la búsqueda, explotando caminos de búsqueda delimitados por una vecindad.

5) Reemplazo de la población, este paso decide si la nueva solución debe ser parte de la población y cuál solución de la población debe ser reemplazada, buscando calidad y diversidad. Una regla de actualización basada en la calidad reemplaza la peor solución de la población, mientras que una regla basada en la diversidad sustituye una solución similar teniendo en cuenta la medida de distancia.

Los Algoritmos Meméticos son hoy el estado del arte en la resolución de diversos problemas de optimización combinatoria, como [62]: problemas de particionado en grafos [47], partición de números [46], conjunto independiente de cardinalidad máxima [45], empaquetado [64], entre muchos otros.

2.1.3 Breve descripción de la lengua nasa

El Nasa Yuwe es la lengua hablada por el pueblo Nasa, que se encuentran ubicados en siete departamentos de la república de Colombia: Cauca, Huila, Tolima, Valle del Cauca, Meta, Caquetá y Putumayo; siendo en el Cauca donde se presenta la mayor población [65]. La interacción con otras comunidades, el mercadeo, las entidades del Estado, las entidades privadas, la iglesia, entre otras, se dio en castellano, llevando a la lengua nasa de la condición de lengua minoritaria a lengua minorizada, debilitándola, negándole espacios de uso y arrinconándola paulatinamente [66]. Actualmente, la lengua nasa es más hablada por los adultos mayores que por los jóvenes o niños, y para algunos su primera lengua es el castellano; a pesar de los esfuerzos realizados por mantener su cultura, su lengua ha sufrido una serie de procesos que han atentado contra su conservación [67].

Inicialmente, la lengua nasa había sido incluida dentro de la Familia Chibcha [68, 69, 70], pero en 1993 Constenla [71] determinó que esta clasificación no es correcta, por lo tanto, fue clasificado como lengua independiente [72] [49]. El nasa yuwe ha sido una lengua de tradición oral, solo hasta el año 2000, se logró unificar el alfabeto nasa, constituyéndose en un mecanismo de identidad que permite a la cultura Nasa ser

fácilmente distinguible [66]. El nasa yuwe es una lengua en proceso de descripción, algunos estudios relevantes sobre esta lengua son: Jung en 1984 [73], CRIC en 2005 [74], Rojas en 1998 [75] y 2009 [76, 49].

La formación de la palabra en nasa yuwe exige la presencia de al menos un radical simple[3] por palabra, el cual deberá aparecer sólo o acompañado de morfemas flexionales[4] o morfemas derivativos[5] [49] [75]. La relación entre tipos de palabra y predicación es importante, las clases de palabra definidas por Tulio Rojas (lingüista experto[6] en varias lenguas indígenas y con más de 40 años de experiencia en el estudio de la lengua nasa yuwe) son [49] [75]:

1. Palabra predicativa:

- Base predicativa con radical lexical, por ejemplo: tulyuth (soy Tulio), memi'kwe (ustedes cantan) y walatha'w (somos grandes).
- Base predicativa con radical gramatical, por ejemplo: personales (idxgu, eres tú), demostrativos (txa', es ese(a/o), deícticos espaciales (ayte', es aquí), interrogativos (madzna', ¿cuánto es?) y cuantificadores (weha', es poco).
- La negación, por ejemplo: thegmeth, no ví y walameg, no eres grande.

2. Nombre. Es la construcción resultante de la aplicación a una base léxica de un conjunto de marcas de flexión, por ejemplo: alku (perro).

3. Palabra Cualificativa. Un radical cualificativo puede entrar en la formación de una palabra predicativa y en la formación de una palabra cualificativa.

4. Conector. Este tipo de palabra no lleva flexión, además no pueden ser bases predicativas. Son empleadas como conectores en la oración. Ejemplos: Sa' (y), atsa' (entonces) y napa (pero).

Es preciso comentar que en la lengua nasa no se encuentra un tipo de palabra igual o similar a los artículos del español o inglés.

[3] Radical simple, es el elemento de base –irreductible- sobre el cual se aplican los diferentes procesos morfológicos para la formación de radicales complejos y para la formación de palabras, por ejemplo: Yat (casa), yat-we'sx (casas) [47].
[4] La flexión se expresa mediante sufijos (flexión modo-personal y la declinación), por ejemplo: alku- (perro), alku-a's (al perro) [47]
[5] La derivación obtiene nuevos radicales, por ejemplo: piya- (aprender), ka-piya-a'h (hacer aprender) [47]
[6] Se puede apreciar el currículo del profesor Tulio Rojas, http://scienti.colciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0000061425

2.2 Trabajos relacionados

A continuación, se presenta una revisión bibliográfica de algunos trabajos sobre los enfoques más relevantes utilizados para la construcción de etiquetadores, construcción de corpus lingüísticos etiquetados y conjuntos de etiquetado. En el Anexo A, se encuentra un detallado estado del arte, el cual se encuentra en revisión para enviarlo a evaluación a una revista internacional.

2.2.1 Etiquetadores basados en información estadística

Los etiquetadores basados en información estadística ofrecen resultados muy competitivos para las diferentes aplicaciones, incluso se utilizan como línea base y para mejorar o iniciar etiquetadores construidos con otras estrategias como redes neuronales, pero son bastante dependientes del tamaño y la calidad del corpus (tanto del entrenamiento como del etiquetado). Entre los trabajos tradicionales (pioneros, pero competitivos a la fecha) se encuentran: En 1996, Ratnaparkhi [29], propone un etiquetador basado en el modelo de Máxima Entropía, MEMM, el cual es una técnica muy flexible y capaz de utilizar diversidad de información contextual y no impone ninguna distribución sobre los datos de entrenamiento. La probabilidad conjunta ($p(h_i, t_i)$) de una historia h y una etiqueta t esta determinada por los parámetros que corresponden a las características activas (α_j, que es $f_j(h,t) = 1$, donde α_j, puede ser un factor de peso efectivo para un predictor contextual en un caso específico). Una característica dada (h,t), puede activarse sobre cualquier palabra o etiqueta en la historia h, y debe codificar cualquier información que puede ayuda a predecir t, tal como la ortografía de la palabra actual, o la identidad de las dos etiquetas anteriores. La palabra específica y el contexto disponible de la etiqueta, se dan en la siguiente definición de historia $h_i = \{w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}\}$. El modelo genera el espacio de características mediante el escaneo de cada par (h_i, t_i) en los datos de entrenamiento. El modelo de prueba utiliza el algoritmo Beam Search (que utiliza la probalidad condicional de la etiqueta ($p(t|h)$), y mantiene arriba los N candidatos con la más alta etiqueta probable en ese punto de la oración. Adicionalmente, cuenta con un procedimiento de búsqueda que opcionalmente consulta un diccionario de etiqueta, el cual tiene una lista de etiquetas para cada palabra del conjunto de entrenamiento. En 2000, Brants [28], propuso un etiquetador de Trigramas, TnT, que utiliza Modelos ocultos de Markov de segundo orden, que también utiliza el algoritmo Beam Search en lugar de Viterbi, reduciendo el tiempo de procesamiento; para sus evaluaciones utilizó el corpus Negra para alemán y el Penn Treebank para el inglés. En 2001,

Lafferty et al [30], presentan un experimento de POST utilizando un modelo de CRF que tienen menor sesgo en el problema de etiquetado, obteniendo menores valores de error comparado con otros.

Los trabajos recientes permiten apreciar las tendencias de estas técnicas en la construcción de etiquetadores, entre ellos, se destacan: En 2003, Mayfield et al [55], proponen un etiquetador basado en grafos que estima las probabilidades conjuntas de transición y emisión usando máquinas de soporte vectorial. La técnica ofrece ventajas sobre otros métodos como la capacidad de incluir características no comunes, soportar un gran número de características y ser neutral con el lenguaje. El artículo presenta la técnica sobre dos aplicaciones de etiquetado como son: reconocimiento del nombre de entidades y etiquetado de palabras, obteniendo buenos resultados en el etiquetado sin tener la necesidad de hacer ajustes específicos para el lenguaje. El enfoque utiliza SVM para estimar las probabilidades de transición del grafo (SVM - Lattice) utilizando el enfoque propuesto por Platt [77]. Los experimentos para POS utilizaron el corpus Penn Treebank, con el 80% de datos para entrenamiento. En 2008, Ekbal y Bandyopadhyay [78], presentan un POST para la lengua bengalí (lengua de la India) utilizando una máquina de soporte vectorial (SVM), el conjunto de etiquetado fue estandarizado en 2006, y consta de 26 etiquetas definidas para lenguas indias. El POST obtuvo una precisión de 80% en contraste con el 56% de la línea base y su desempeño fue correcto con las palabras desconocidas. En 2013, Ismael, et.al, [79] presentan la construcción de un diccionario etiquetado para partes del discurso de la lengua Bangla a través de la propuesta de un algoritmo para esta tarea. Una lista de sufijos fue creada manual mente. El algoritmo fue evaluado usando un parágrafo etiquetado manualmente con 10000 palabras y 11 etiquetas, se encontró que el algoritmo propuesto tiene más precisión para verbos que para nombres y adjetivos. El diccionario de etiquetado para POS construído puede ser usado para categorización de palabras, desambiguación de palabras, análisis morfológico de verbos y nombres y para análisis de la etiqueta más adecuada para la lengua Bangla. Se recopilaron 320.443 palabras de la lengua Bangla recolectadas de periódicos, blogs, y otros sitios web. En 2014, Kardan y Bahojb [80], presentan un POST para lengua Persa, el cual incluye dos pasos: 1) Extracción de características del idioma y 2) POS Tagging, utilizando el método de Máxima Entropía (MEM), se compara con versiones anteriores para esta lengua como son: un etiquetador basado en memoria (Memory Based

Tagger, MBT) [81][7] y una estimación por máxima probabilidad (Maximum Likelihood Estimation, MLE)[8] [81], en donde, el POST propuesto obtuvo una precisión mayor con un valor de 97.81%. En 2014, Ariaratnam et al [36], presentan un analizador sintáctico sencillo (shallow parser) para la lengua Tamil (lengua del Sur de India y Sri Lanka), cuyo primer paso es la construcción de un etiquetador basado en MEMM, que utilizó un corpus pequeño y de libre acceso, con 20 etiquetas y etiquetado manual; utilizó 10000 palabras para entrenamiento y 2500 para pruebas. También en 2014, Makazhanov et al [33], presentan un etiquetador para el Kazakh, que utiliza dos etiquetadores estadísticos (MEMM y HMM combinado con SVM). En 2016, Albared et al [82], proponen seis experimentos con diferentes modelos léxicos para etiquetar palabras desconocidas para el árabe que incluyen el uso de un analizador morfológico y la estimación de sufijos y prefijos para las palabras desconocidas, sumado al uso de interpolación lineal propuesto por Samuelsson [83]. En 2017, Keyaki y Miyazaki [84], presentan un método de POST para Web search queries, que propone tres métodos para determinar las etiquetas del POS: Máxima Frecuencia, Máxima probabilidad y la combinación de estos dos.

2.2.2 Etiquetadores basados en reglas

Los etiquetadores basados en reglas son un poco menos complejos que los estadísticos, pero construir reglas es costoso y dispendioso e implica un mayor conocimiento del lenguaje. Existen dos trabajos relevantes tradicionales propuestos por Brill. El primero en 1992 [85], en el cual busca automatizar el proceso de etiquetado basado en reglas, obteniendo un etiquetador robusto, que plantea una reducción en la información requerida, mayor claridad en un pequeño conjunto de reglas significativas en comparación con las grandes tablas de los etiquetadores estadísticos. Utilizó el Corpus Brown para las evaluaciones. En 1995, Brill [31], presenta un etiquetador basado en transformaciones direccionadas por el aprendizaje del error, el cual obtiene valores de precisión superiores a los de un etiquetador basado en modelos ocultos de Markov. Los experimentos utilizaron tres Corpus Pen Treebank Tagged, Corpus Brown y Pen Brown. Otros trabajos recientes y relevantes incluyen los siguientes: En 2015, Alsuhaibani et al [86], presentan un etiquetador especializado para código fuente y comentarios, dado que la mayoría de los etiquetadores funcionan muy bien cuando existe una estructura en la escritura de las oraciones a evaluar, pero para el caso de los comentarios en código fuente no funcionan adecuadamente. También en 2015,

[7] MBT, tiene en cuenta la etiqueta de las anteriores palabras tanto para palabras conocidas como desconocidas [78]
[8] MLE, orientado a seleccionar la etiqueta que tenga la máxima probabilidad para esa palabra [78]

Azis y Sunitha [87], proponen un etiquetador hibrido para la lengua malayalam, que combina las reglas tradicionales con los n-gramas, el cual produce mejores resultados comparado con otras estrategias como HMM y TnT.

2.2.3 Etiquetadores basados en redes neuronales

Estos etiquetadores requieren mayor tiempo de entrenamiento, y al igual que los estadísticos y los basados en reglas, necesitan alta calidad en los datos y mayor conocimiento de la lengua. En la literatura, se encuentran trabajos relevantes así: En 1994, Schmid [32], basándose en el trabajo de Nakamura and Shikano (1989) [88], utiliza un perceptrón multicapa para realizar el etiquetador, alcanzando valores de precisión muy buenos, comparables con etiquetadores de la época basados en Trigrams y HMM pero con baja velocidad de procesamiento. En sus pruebas utilizó el Corpus Pen Treebank para el inglés. En 2001, Pérez y Forcada [89], utilizan redes neuronales recurrentes para construir un etiquetador, el cual no requiere etiquetado manual para el entrenamiento, obteniendo resultados similares a HHM, utilizó el corpus Penn Treebank. En 2016, Kabir et al [90], presentan una propuesta de POST para la lengua bengalí, que utiliza el corpus desarrollado por Microsoft Research India [91] y utiliza un POST que consta de dos partes, en primera instancia, Construcción del Vector de Características (Feature Vector Construction), y en segunda instancia, un clasificador que utiliza arquitectura de red profunda (Deep Belief Network) de tres capas; para el experimento se utiliza validación cruzada de 10-folders, obteniendo valores de recuerdo de 93.33% y quedando pendiente las comparaciones con otras investigaciones. En 2017, Hnin et al [92], presentan un modelo de POST basado red neuronal de propagación hacia atrás (Back-propagation) para la lengua Myanmar, entrenándolo con 3-grama, 4-grama y 5-grama, obteniendo altos resultados, se utilizó un conjunto de datos (data set) de prueba abierto, desempeñándose mejor que HMM con reglas.

2.2.4 Identificadores construidos mediante proyección de etiquetas

Entre los trabajos relevantes para la construcción de identificadores de partes del discurso que utilizan proyección de etiquetas, tanto para un lenguaje como para múltiples lenguajes, se encuentran: En 2015, Zennaki et al [93], proponen un POST para lenguajes que no tienen muchos recursos, basado en proyección entre lenguajes (cross-languages) de anotaciones lingüísticas de corpus paralelos. El POST utiliza redes neuronales recurrentes (RNN) para la alineación de palabras y TnT [28] como POST; utilizan las 12 categorías (tagset) propuestas en [10], para el etiquetado de las palabras en los corpus. En 2014, Duong et al [94], proponen un POST para lenguas

pobres en recursos desde un enfoque multilenguaje, utiliza un etiquetador de enfoque supervisado de Máxima Entropía para hacer el etiquetado del lenguaje origen o rico (inglés) luego mediante alineación paralela de palabras hace la proyección de etiquetas al lenguaje pobre (8 lenguas europeas).

2.2.5 Identificadores construidos utilizando algoritmos metaheurísticos

En los últimos años, se ha venido presentado el uso de algoritmos metaheurísticos para resolver el problema de etiquetado, obteniendo resultados comparables con el estado del arte usando los enfoques tradicionales. A continuación, se presenta una breve descripción de la revisión exhaustiva que se realizó en esta área. Los trabajos encontrados, agrupados por autor son:

- En 2010, Lv et al [95], presentan un etiquetador basado en programación de expresión genética (genetic expression programming, GEP), el cual es capaz de aprender de un corpus de entrenamiento, los cromosomas son generados desde las tablas de entrenamiento. Cada gen está compuesto de una etiqueta e información contextual útil para el etiquetado de una palabra dentro de una oración. El algoritmo de etiquetado GEP propuesto involucra: la inicialización de la población, a través de una tabla mixta uniforme según referencias citadas en el documento [96], la construcción de una tabla estándar generada aleatoriamente según el método propuesto también en [96] y ajustarla mediante el mismo método. Se utiliza el corpus Brown, para los experimentos dejando 100.000 palabras para test, obteniendo los mejores resultados de precisión para el etiquetador GEP propuesto (97.40%), en contraste con etiquetadores basados en algoritmos genéticos, redes neuronales y HMM. Adicionalmente, realiza experimentos que resaltan el impacto del tamaño del corpus en el desempeño de los etiquetadores. También en 2017, Lv et al [97] en 2017, presenta un etiquetador llamado Programación de expresión genética de diseño uniforme (uniform-design genetic expression programming, UGEP), en el cual se utilizan dos corpus (Brown Corpus y WSJ Corpus) y se evalúa el desempeño frente a etiquetadores de algoritmo genético [95], redes neuronales y HMM en dos experimentos uno con un léxico cerrado (los datos de entrenamiento tienen todas las palabras de los datos de prueba) y otro con léxico abierto (no todas las palabras aparecen en el corpus de entrenamiento), obteniendo mejores valores de precisión de UGEP para el caso del léxico cerrado (Brown Corpus 98.37% y WSJ Corpus 98.83%), que para el léxico abierto (Brown Corpus 97.4% para las palabras conocidas, 88.6% para palabras

desconocidas y 96.52% para todo el sistema), en todos los casos UGEP supera los otros etiquetadores. En el trabajo, se destaca como inconveniente de la propuesta la baja velocidad en el procesamiento en comparación con los métodos estadísticos, al igual que el impacto del tamaño del corpus en el desempeño de los etiquetadores.

- En 2010, Forsati et al [98]**,** presentan un etiquetador basado en la metaheurística de la búsqueda armónica, llamado HSTAGer, el cual obtiene resultados comparables con enfoques estadísticos demostrando efectividad y robustez. Las evaluaciones se realizaron con el corpus Brown y se obtuvieron resultados de alta calidad en comparación con etiquetadores basados en algoritmos genéticos y recocido simulado (para la época de la propuesta). Los autores también sostienen que el algoritmo propuesto es independiente del lenguaje, aunque se evalúa con inglés americano. En 2012, Forsati y Shamsfard [41], presenta un etiquetador denominado BEETAGer modelado desde la perspectiva del enfoque de cooperación entre un algoritmo de optimización de colonia de abejas y la evaluación de individuos basada en una tabla de entrenamiento extraída del corpus etiquetado de Brown. La función objetivo del algoritmo de optimización evolutivo es simplificada sustancialmente mediante los modelos de etiquetado estadístico. Los resultados muestran que el algoritmo propuesto obtiene mejores soluciones con alta calidad considerando las medidas de precisión de otros algoritmos conocidos.

- En 2014, Forsati y Shamsfard [99], utilizan los algoritmos de búsqueda armónica y optimización de colonia de abejas (BCO), para modelar el problema de etiquetado. Las tablas de información estadística son tomadas del corpus Brown, y luego se utilizan HS y BCO para proponer los etiquetadores HSTAGer y BEETAGer. Los experimentos realizados muestran que se obtienen mejores resultados con BEETAGer. En 2015, Forsati y Shamsfard [19], proponen una nueva versión del etiquetador llamado HSTAGer, el cual parte desde un enfoque estadístico como HMM para la estimación de probabilidades y luego hace la adaptación del problema de etiquetado con el algoritmo de optimización de búsqueda armónica (Harmony Search, HS), para los experimentos utilizan tres datasets dos tomados del corpus Brown y uno del Corpus Penn Treebank, se configuran de manera empírica los parámetros de HS (cambiar estos parámetros impacta significativamente la precisión), se evalúan varias funciones de optimización para el problema,

obteniendo resultados comparables y mejores para los diferentes datasets tanto en términos conocidos como en los desconocidos.

- En 2012, Silva et al [26] modelan el problema de etiquetado como un problema de optimización combinatoria, en donde utilizan el algoritmo para la evolución de un conjunto de reglas de desambiguación, que luego son usadas como una heurística para guiar la búsqueda para la mejor combinación de etiquetas, utilizaron NLTK para construir el etiquetador llamado GA-Tagger, utilizaron el corpus Brown y 10% del Corpus Penn Treebank haciendo evaluaciones con diferentes poblaciones y generaciones y comparando la experiencia con otras soluciones que incluyeron el uso de Metaheurísticas y técnicas tradicionales como la propuesta por Brill [31], obteniendo muy buenos resultados. También en 2012, Silva et al [100], presentan un enfoque que utiliza tanto las ventajas de los etiquetadores basados en estadística como en reglas, tomando el problema de etiquetado como un problema de clasificación, donde las clases son las diferentes partes del discurso y los atributos predictivos son la información de contexto y algunos aspectos sobre la estructura interna de las palabras. De tal forma, que el sistema propuesto se compone de dos pasos: primero, unas reglas de desambiguación son descubiertas utilizando un algoritmo evolutivo y luego un etiquetador evolutivo etiqueta las palabras utilizando las reglas encontradas, la función objetivo del segundo algoritmo evolutivo se basa en datos estadísticos. Obtienen valores de precisión competitivos con otras experiencias con las que se comparan. En 2013, Silva et al (2013) [43], modelan el problema de etiquetado desde una nueva aproximación dividiéndolo en 2 tareas: aprendizaje y optimización, a través del uso de algoritmos genéticos y el algoritmo PSO, también es basado en reglas y genera una función de optimización diferente, se compara con GA-Tagger (un trabajo previo de los mismos autores) y otros trabajos, obteniendo una mejora en relación con el estado del arte y con GA-Tagger. También en 2013, Silva et al [18], presentan el uso del algoritmo de inteligencia de enjambre (PSO) para el problema de etiquetado, usando como modelo de información un conjunto de reglas de desambiguación extraídas previamente mediante un algoritmo evolutivo, el problema de búsqueda para la mejor etiqueta es visto como un problema de optimización combinatoria, donde la solución es evaluada con la ayuda de las reglas anteriormente aprendidas. Los resultados mostraron una mejora en los valores de precisión en comparación con las experiencias previas [26, 100]. En 2014, Silva et al [101], presentan una

estrategia de división del problema de etiquetado con las mismas tareas de [43], los supuestos de los que se parte en este trabajo son: con la ayuda de un algoritmo de clasificación es posible generalizar recursos lingüísticos y la información típicamente usada en los enfoques probabilísticos, mediante el aprendizaje de un conjunto de reglas de desambiguación, las cuales son usadas como una heurística para ayudar a solucionar la tarea. El segundo supuesto, es que es posible modelar el problema principal como un problema de búsqueda y usar las reglas descubiertas en la primera fase como una heurística para orientar la búsqueda en el espacio de solución del problema. Los autores lograron una mejora marginal en los valores de precisión obtenidos en anteriores investigaciones [26, 43, 100].

- En 2014, Bachir [102], presenta el uso de algoritmos genéticos para modelar y solucionar el problema de desambiguación lingüística, aplicado a un corpus grande (MSA) de la lengua árabe, comparándolo con otros métodos existentes, obteniendo buenos resultados, aunque no fue posible hacer comparaciones con otros métodos dado que los datos eran diferentes.

- En 2011, Ekbal y Saha [103], proponen un clasificador basado en un enfoque multiobjetivo, caracterizado por que las características son seleccionadas y evaluadas en su mayoría sin usar un profundo conocimiento de lenguaje, se evaluó en tres lenguas pobres de recursos (bengalí, hindi, y telugu) obteniendo valores buenos de precisión, recuerdo y medida F. En 2013, Ekbal y Saha [42], presentan un etiquetador dentro de una estructura de técnicas de optimización mono y multiobjetivo, primero se utiliza un clasificador mono-objetivo que explota la capacidad de búsqueda del algoritmo de recocido simulado (Simulated Annealing), luego se utiliza el enfoque multiobjetivo para solucionar el mismo problema utilizando AMOSA (técnica basada en simulado recocido). Se evaluaron los resultados con dos lenguas indias (bengalí e hindi) obteniendo mejores valores con la versión multiobjetivo.

- En 2002, Lourdes [104], presenta un etiquetador basado en un algoritmo genético, en el cual los individuos son las secuencias de etiquetas asignadas a las palabras en una oración, los datos están organizados como contextos (es decir, que tienen en cuenta las etiquetas que tienen las predecesoras y sucesoras de la palabra a etiquetar) y para incrementar la precisión del algoritmo se varía la longitud de los

contextos en la función fitness. Para los experimentos se utilizó el corpus Brown. La precisión obtenida fue comparable con otros enfoques (Brill y HMM) aunque no se presentan los detalles de las comparaciones y de los resultados obtenidos, también resalta los puntos en los que más frecuentemente se incurren en errores en el etiquetado. En 2006, Lourdes et al [40], utilizan tres algoritmos diferentes para abordar el problema de etiquetado: un genético clásico, CHC (variación de un genético que promueve diversidad), y recocido simulado. Para los experimentos utilizaron dos corpus, Brown y Susanne (ambos para el inglés), se hicieron varias pruebas para cada algoritmo y se contrastaron con enfoques como el método Viterbi obteniendo resultados comparables y en algunos casos los algoritmos propuestos fueron mejores.

- En 2005, Wilson y Heywood [105], describen una variación del modelo de Brill [31] al utilizar un algoritmo genético para generar la instanciación de las reglas y proveer un orden adaptativo a través de un proceso natural mediante el operador de cruce, obteniendo valores de precisión competitivos pero más bajos que los obtenidos con el modelo de Brill y HMM, sin embargo, se demuestra su eficiencia para el problema de etiquetado.

2.2.6 Conjunto de etiquetas para etiquetar Corpus Lingüísticos

A continuación, se presentan los trabajos relacionados con el conjunto de etiquetas a utilizar en el etiquetado de un corpus:

En 2008, Rabbi, et.al, [106] presentan el procedimiento seguido para el diseño de un tagset para Pashto Language, teniendo en cuenta las EAGLES guidelines for morphosyntactic annotation of corpora [107], obteniendo 215 Tags distribuidos así: 26 etiquetas para Nombre (*Noun*), 77 para Verbo (*Verb*), 60 para Pronombres (*Pronouns*), 19 para Adjetivos (*Adjectives*), 15 para Puntuación (*Punctuation*), 7 para Adverbio para (*Adverb*), 3 para preposiciones y posposiciones (*Adposition*), 6 para palabras extranjeras (*for Foreign Words*) y 1 para Interjección y conjunción (*Interjection and Conjunction*). También en 2008, Baskaran, et.al, [91], presentan IL-POSTS un framework que contiene un conjunto de etiquetas para la mayoría de las lenguas indígenas, teniendo en cuenta EAGLES guidelines [107], el cual tiene como propósito ser de uso general. En este trabajo se describen las características del diseño metodológico y las estrategias metodológicas que dan origen al Framework. En 2012, Petrov, et.al, [10] presentan un conjunto de 12 etiquetas unificadas a partir de 25

conjuntos de etiquetas para 25 lenguajes provenientes de trabajos previos, la propuesta busca mejorar la precisión de los identificadores de partes del discurso a través de varios lenguajes. Los 12 POS Tags definidos son: NOUN (*nouns*), VERB (*verbs*), ADJ (*adjectives*), ADV (*adverbs*), PRON (*pronouns*), DET (*determiners and articles*), ADP (*prepositions and postpositions*), NUM (*numerals*), CONJ (*conjunctions*), PRT (*particles*), '.' (*punctuation*) and X *(a catch-all for other categories such as abbreviations or foreign words*). Es una propuesta que ha sido muy aceptada y ampliamente utilizada en los diferentes trabajos de POS Tagging, como por ejemplo en el proyecto NTLK [15], en donde la utilizan con los diferentes corpus y etiquetadores y ofrecen las equivalencia de las etiquetas del corpus como Brown [108] y PennTreeBank [109] con las de este conjunto de etiquetado, esto dos corpus son ampliamente utilizados para analizar y comparar desempeño de los diferentes etiquetadores propuestos. En 2014, Dinakaramani, et.al, [110] establecieron un conjunto de 23 POS Tags para etiquetar 10.000 oraciones del corpus IDENTIC de la lengua Indonesia, que contenían 262.330 tokens, definieron tres principios para el conjunto de etiquetas (linguistically valuable, simplicity, automatically refinable) y una Metodología para etiquetar manualmente (utilizaron dos anotadores humanos) el corpus con el Tagset propuesto.

2.2.7 Corpus Lingüísticos etiquetados para identificadores de partes del discurso

Un corpus lingüístico es una parte vital en el procesamiento de lenguaje natural (NLP), de tal forma que, con el fin de establecer las características principales y elementos que constituyen los corpus y los diferentes métodos de etiquetado se han revisado varios trabajos tanto para lenguas tradicionales (inglés, francés, portugués, entre otras) como para lenguas no tradicionales. Construir un corpus etiquetado, así como su correspondiente conjunto de etiquetas es crucial para el procesamiento de lenguaje natural, especialmente para la identificación de partes del discurso. A continuación, se presentan algunos trabajos relacionados:

En 1979, Francis y Kucera [108] propusieron el Corpus Brown, para inglés americano conteniendo aproximadamente 1.014.312 palabras en categorías de textos (como reportajes, editoriales, reviews, entre muchas otras), este corpus es muy utilizado. Ha sido ampliado en varias ocasiones, actualmente cuenta en total con 473 categorías provenientes de las subdivisiones de los 82 tags principales y es muy utilizado para el etiquetado en inglés. En 1993, Marcus et.al, [109] presentan el corpus Penn Treebank con una reducción en el tagset en comparación con el tagset del corpus Brown, y

teniendo en cuenta el contexto sintáctico de la palabra a etiquetar, de tal forma, que se obtuvieron 48 Tags. El proceso de etiquetado fue automático y con corrección manual. El corpus consiste de cerca de 4 millones de palabras de inglés americano (World Street Journals) y es muy utilizado para tareas de POS Tagging. En 2005, Kohen [111] presenta el Corpus Europarl extraído de los Proceedings del Parlamento Europeo, que incluye versiones en la mayoría de las lenguas europeas. Este corpus inicialmente fue construido pare ser utilizado en tareas de traducción automática (machine translation), señala 5 pasos para su recopilación (obtención de documentos en varios idiomas del sitio web del Parlamento Europeo, dividir el texto en oraciones, preparar el corpus para su uso, mapear sentencias en los lenguajes). Este corpus ha sido muy utilizado en tareas de POS Tagging para multilenguaje. En 2010, Ahmed y Qadir [112] describen el análisis que se hizo para definir el Tagset para el Shindi, su aplicación en el etiquetado de las palabras, así como los problemas que se presentaron al aplicarlo. En 2012, Spoustová y Spousta [113] presentan el proceso de construcción de un gran corpus de Checo, que involucró en primera instancia, una revisión y limpieza manual de los documentos duplicados de los sitios, en segunda instancia se utilizó un algoritmo de casi duplicado (near-duplicate algorithm) para remover párrafos duplicados de los documentos utilizando una medida de similitud basada en una comparación de n-gramas, en tercera instancia, se desarrolló un módulo de detección de lenguaje (Language detection module) para eliminar palabras provenientes del Slovak, que consta de dos filtros unaccented words y general language. El corpus contiene 2.65 billones de palabras provenientes de artículos de noticias y revistas, 1 billón de palabras de blogs, periódicos y otras unidade literarias no revisadas, 1.1 billón de palabras de mensajes de discusión, resaltando la alta calidad de las palabras del corpus dada la intervención humana en el proceso de construcción del corpus. En 2014, Sing and Banergee [114] presentan el etiquetado de un corpus para la lengua Bhojpuri (lengua del norte de India), que utiliza el BIS scheme, definido en 2010, los datos del corpus corresponden a 9 historias populares con aproximadamente 5300 palabras etiquetadas de 3 historias, los datos fueron recopilados de conversaciones y luego fueron transcritos. El tagset incluye 33 categorías, las cuales tienen subniveles. En el trabajo se presentan las características del idioma, observables a la luz del etiquetado. En 2014, Ariaratnam, et.al, [36] describen el proceso de etiquetado de 500.000 palabras recolectadas de periódicos de Sri Lankan Tamil, dado que no existe un corpus disponible para el Tamil; entre los pasos seguidos se encuentran en primera instancia, el pre-procesamiento, en donde se extrajeron las oraciones con 20 o menos

palabras para facilitar el proceso y se hizo una pre-edición del corpus para corregir errores de escritura y eliminar espacios innecesarios, en segunda instancia, se propuso un conjunto de 20 etiquetas con el apoyo de un lingüista, en tercera instancia, se realizó el etiquetado manual creando un corpus etiquetado de 12.500 palabras, y se hizo la debida revisión al etiquetado. También en 2014, Scherrer, et.al, [115] presentan un gran corpus multilingual para Alemán, Francés, Italiano, e Inglés, en donde se aprecia el procesamiento y el etiquetado automático de los textos provenientes de archivos HTML, utiliza el etiquetado Universal propuesto en [10] para la descripción de las palabras, la evaluación se hizo manual en pequeños fragmentos del corpus. El corpus cuenta con más de 6 millones de palabras para cada lengua.

2.3 Análisis comparativo de la literatura revisada

En las **Tabla 2.4** a **Tabla 2.9** se puede apreciar un análisis comparativo sobre los trabajos relevantes presentados y otros adicionales que se encuentran en el Anexo A.

2.3.1 Enfoques de construcción de identificadores de partes del discurso

Etiquetadores basados en información estadística

Autor(es)	Año	Lengua etiquetada	Técnica Base	Corpus utilizado	Aspectos a destacar
Ratnaparkhi [29]	1996	Inglés	Modelo de Máxima Entropía	Penn Treebank	• Ofrecen resultados muy competitivos para las diferentes aplicaciones, incluso actualmente se utilizan para mejorar o iniciar etiquetadores construidos con otras estrategias como redes neuronales. • Son bastante dependientes del tamaño y calidad del corpus (tanto para datos de entrenamiento como etiquetado) • Se encuentran diferentes propuestas para construir POST para lenguas diferentes a las tradicionales. • Presentan comportamiento impredecible dado la dependencia de los datos.
Brants [28]	2000	Inglés	Trigramas TnT (Modelos Oculto de Markov de segundo orden	Corpus Negra para alemán y el Penn Treebank para el inglés	
Lafferty et al [30]	2001	Inglés	CRF	Penn Treebank	
Mayfield et. al	2003	Inglés	SVM	Penn Treebank	
Ekbal y Bandyopadhyay [78]	2008	Bengalí	SVM	Corpus con 72,341 palabras de diferentes tipos, etiquetadas con 26 etiquetas	
Ismael et. al	2013	Bangla	Algoritmo propuesto para etiquetar palabras	Corpus etiquetado manualmenete con 10,000 palabras y 11 etiquetas	
Kardan y Bahojb [80]	2014	Persa	Extracción de características y Modelo de Máximo Entropía	Persian Corpus	
Bach et al [116]	2018	Vietnames	CRF	Vietnames Facebook	

Tabla 2.1. Análisis comparativo sobre identificadores de partes del discurso enfoque estadístico. Parte I

Autor(es)	Año	Lengua etiquetada	Técnica Base	Corpus utilizado	Aspectos a destacar
Ariaratnam et al [36]	2014	Tamil	MEMM	Corpus pequeño etiquetado manual mente con 20 etiquetaas	• La mayoría de los etiquetadores propuestos trabajan para la lengua inglés, obteniendo muy buenos resultados sobre corpus grandes y muy bien etiquetados, pero para corpus diferentes que no poseen la riqueza de recursos del inglés, no se obtienen los mismos resultados. • Disponibilidad de recursos lingüísticos sobre los cuales pueden trabajar para el caso de lenguas tradicionales, como: grandes corpus etiquetados, corpus paralelos y diccionarios.
Makazhanov et al [33]	2014	Kazakh	MEMM y HMM con SVM	Kazakh Language Corpus	
Paul et al [53]	2015	Nepalí	HMM	Nepalí text Corpus	
Jiancho [117],	2015	Inglés	MEMM	Inglés	
Ranjan et al [118]	2015	Odia	SVM	Odia corpus con 10,000 palabras y 5 etiquetas	
Zhonglin et al [119]	2016	China	Diccionario segmentado para palabras sin ambigüedades y Máxima Entropía del analizador Standford para palabras ambiguas	People's dayly. Corpus etiquetado manualmente	
Sun y Wan [120]	2016	China			
Albared et al [82]	2016	Árabe	HMM	29340 palabras etiquetadas manualment de dos tipos de texotos arabes.	
Keyaki y Miyazaki [84]	2017	Inglés	Máxima Frecuencia Máxima Probabilidad Combinación	ClueWeb09 Category B	

Tabla 2.2. Análisis comparativo sobre identificadores de partes del discurso enfoque estadístico. Parte II

Etiquetadores Basados en reglas

Autor(es)	Año	Lengua etiquetada	Técnica Base	Corpus utilizado	Aspectos a destacar
Brill [85]	1992	Inglés	Basado en reglas	Brown Corpus	• Requieren un amplio conocimiento sobre el lenguaje, por tanto, el proceso de construir reglas es costoso y dispendioso. • Existen trabajos para diferentes aplicaciones. • Presentan muy buen desempeño
Brill [31]	1995	Inglés	Transformation-Based Error-Driven Learning	Peen Treebank, Corpus Brown y Pen Brown	
Alsuhaibani et al [86]	2015	Inglés	Heurística y un programa estático de análisis de información	Códigos fuente	
Azis y Sunitha [87]	2015	Malayalam	Reglas tradicionales y bigramas	Malayalam Corpus con 39 etiquetas	
Mall y Jaiswal [121]	2015	Hindú	5 reglas lingüisticas para la identificación de partes del discurso	IIT-Hyderabad	
Tian y Lo [122]	2015	Inglés	Unigram, TnT, Tree, NLTK, Standford, TBT, Annie	Bug Report Collection	

Tabla 2.3. Análisis comparativo identificadores de partes del discurso basados en reglas

Etiquetadores basados en Redes Neuronales

Autor(es)	Año	Idioma	Técnica Base	Corpus utilizado	Aspectos a destacar
Nakamura and Shikano [88]	1989	Inglés	N-gram Network	Corpus Brown	
Schmid [32]	1994	Inglés	Red perceptron multicapa	Corpus Penn Treebank	
Pérez y Forcada [89]	2001	Inglés	Red Neuronal Recurrente	Corpus Penn Treebank	
Poe et al [123]	2008	Holandés	Red neuronal	Corpus Gesproken Nederlands con 72 etiquetas	• En algunos casos requieren conocimientos específicos del lenguaje para el afinamiento de los parámetros.
Carneiro et al [124]	2010	Portugués	Red neuronal artifical sin pesos	CETENFolha	• Se encuentran propuestas que trabajan el problema de etiquetado desde una perspectiva multilenguaje
Carneiro et al [35]	2015	8 lenguajes	Multilingual Red neuronal artifical sin pesos	Mandarín, inglés, japonés, portugués, italiano, alemán, ruso, turco	
Kabir et al [90]	2016	Bengalí	Red profunda de tres capas	IL-POST	
Hnin et al [92]	2017	Myanmar	Red neuronal de propagación hacia atrás	Myanmar corpus etiquetado manualmente	

Tabla 2.4. Análisis comparativo identificadores de partes del discurso basados en redes neuronales

Etiquetadores basados en Proyección de etiquetas

Autor(es)	Año	Idioma	Técnica Base	Corpus utilizado	Aspectos a destacar
Das y Petrov [34]	2011	Bilingüe	Propagación de etiqueta basada en grafos	Monolingual treebanks y textos paralelos con inglés	• Alineación de corpus desde una lengua rica en recursos con una lengua que no tenga recursos, pero si tenga traducciones a la lengua rica. • Desajustes entre los etiquetados usados para los lenguajes (en los casos multilenguaje) • Corpus alineados con ruido y divergencia sintáctica (diferencia del orden de las secuencias correctas) en varios idiomas. • En su mayoría, utilizan enfoques supervisados (generalmente estadísticos) para la construcción del etiquetador
Duong et al [37]	2013	Bilingüe	Proyección de etiquetas	Corpus paralelo Europarl	
Duong et al [94]	2014	multilenguaje	Etiqueta el recurso usando un etiquetador supervisado y luego se usa alineación de palabras	CoNLL (Treebanks de 13 lenguajes) mapeados con las 12 etiquetas Universales.	
Zennaki et al [93]	2015	Multilenguaje	Redes neuronales recurrentes combinada con un sistema básico de proyección multilenguaje.	ARCADE II y Europarl CoNLL	

Tabla 2.5. Análisis comparativo sobre identificadores de partes del discurso basados en proyección de etiquetas

Etiquetadores basados en algoritmos Metaheurísticos

Autor(es)	Año	Lengua etiquetada	Técnica Base	Corpus utilizado	Aspectos a destacar
Lourdes [104]	2002	Inglés	Algoritmo genético	Corpus Brown	• Proveen una opción eficiente y robusta para el problema de etiquetado. • Los algoritmos metaheurísticos alcanzan desempeños muy cercanos a los mejores con poblaciones pequeñas y pocas iteraciones. • En los trabajos revisados se puede apreciar que con una pequeña mejora (cambio, adición) a la anterior propuesta, se obtiene una mejora en el desempeño del etiquetador. • Para iniciar su proceso de construcción parten de enfoques estadísticos o basados en reglas para la construcción del POS Tagging, desde una perspectiva más sencilla. • La mayoría de las investigaciones son para el inglés.
Wilson y Heywood [105]	2005	Inglés	Algoritmo genético	Penn Treebank	
Lourdes et al [40]	2006	Inglés	Algoritmo genético (CHC) y Recocido simulado	Corpus Brown y Susanne	
Forsati et al [98]	2010	Inglés	Búsqueda armónica	Corpus Brown	
Lv et al [95]	2010	Inglés	Programación de expresión genética	Corpus Brown	
Ekbal y Saha [103]	2011	Bengalí, hindi, y Telugu	Recocido simulado multiobjectivo	Bengali news corpus	
Forsati y Shamsfard [41]	2012	Inglés	Algoritmo de optimización Colonia de abejas	Corpus Brown	
Silva et al [26]	2012	Inglés	Algoritmo genético	Corpus Brown y Penn Treebank	
Silva et al [100]	2012	Inglés	Algoritmo evolutivo para generar reglas y etiquetador evolutivo	Corpus Brown y Penn Treebank	

Tabla 2.6. Análisis comparativo sobre identificadores de partes del discurso basados en algoritmos Metaheurísticos. Parte I

Autor(es)	Año	Lengua etiquetada	Técnica Base	Corpus utilizado	Aspectos a destacar
Ekbal y Saha [42]	2013	Bengalí e hindi	Dos clasificadores ensamblados, el primero uno monoobjetivo y el segundo multiobjetivo ambos basados en recocido simulado.	Para bengalí NLPAI ML Contest-2006 (27 etiquetas) and SPSAL-2007 (26 etiquetas). Para Hindi de SPSAL-2007	
Silva et al [43]	2013	Inglés	Algoritmo genético y PSO	Corpus Brown y Penn Treebank	
Silva et al [18]	2013	Inglés	Inteligencia de enjambres (PSO)	Corpus Brown y Penn Treebank	
Forsati y Shamsfard [99]	2014	Inglés	Búsqueda armónica y Colonia de abejas	Corpus Brown	
Silva et al [101]	2014	Inglés	Algoritmo evolutivo y PSO	Corpus Brown y Penn Treebank	
Forsati y Shamsfard [19]	2015	Inglés	Búsqueda armónica	Corpus Brown y Penn Treebank	
Lv et al [97]	2017	Inglés	Programación de expresión genética de diseño uniforme	Corpus Brown y Penn Treebank	

Tabla 2.7. Análisis comparativo sobre identificadores de partes del discurso basados en algoritmos Metaheurísticos. Parte II

Algunas brechas encontradas en la revisión presentadas en las **Tabla 2.4** a **Tabla 2.7** son:

- En la mayoría de los casos no se explica cómo se replican las experiencias de cada trabajo con los que se compara y cómo se configura la propia propuesta. Esto dificulta la replicabilidad de los experimentos.

- Se pueden presentar diferencias en los valores de precisión de los algoritmos al momento de replicar los experimentos, a pesar de que un corpus etiquetado se encuentre disponible, el desconocimiento sobre cómo se procesó el corpus y cómo fue utilizado causa una disminución significativa en la precisión.

- A pesar de las diferentes propuestas, todavía es necesario buscar soluciones para el etiquetado que sean menos complejas y usables en diferentes lenguas.

- Los trabajos revisados sobre etiquetadores que utilizan algoritmos metaheurísticos son recientes y como se puede apreciar son pocas la metaheurísticas se han sido evaluadas en este problema especifico.

- Los algoritmos meméticos no se han utilizado en el problema de etiquetado y son el estado del arte actual en diversos problemas complejos de optimización combinatoria [45, 46, 47, 62, 64].

2.3.2 Corpus lingüísticos etiquetados y conjuntos de etiquetas

A continuación, en la **Corpus lingüísticos etiquetados**

Autor(es)	Año	Idioma	Aspectos a resaltar de la revisión realizada
Francis y Kucera	1979	Inglés	• Construir un corpus lingüístico no es una tarea fácil, que toma tiempo y es costosa dado lo dispendioso de la tarea. • No resuelven problemas para lenguas no tradicionales que poseen poca descripción, información y los datos de los corpus son pequeños • Para lenguas tradicionales (europeas e inglés americano) ya existen corpus muy utilizados con excelentes características como: tamaño,
Marcus et al	1993	Inglés	
Kohen	2005	Multilenguaje (lenguas comunidad europea)	
Ahmed y Qadir	2010	Shindi	
Spoustová y Spousta	2012	Checo	
Ariaratnam, et al	2014	Tamil	
Sing and Banergee	2014	Bhojpuri	

Autor(es)	Año	Idioma	Aspectos a resaltar de la revisión realizada
Scherrer, et al	2014	Multilingual (alemán, francés, italiano, e inglés)	variedad, han sido varias veces revisados, es decir, son confiables. • Para el caso de lenguajes pobres de recursos lingüísticos, la realización de estos corpus es aún más costosa porque en la mayoría de los casos hay que hacer el etiquetado manual, dado que no hay ninguno o son escasos los corpus (o textos en la lengua) • De la revisión realizada se puede obtener una imagen del contexto metodológico para la creación de un corpus etiquetado.

Tabla 2.8 y **Tabla 2.9** se presentan algunos aspectos a resaltar sobre corpus lingüísticos etiquetados y conjuntos de etiquetas en relación a los trabajos revisados.

Corpus lingüísticos etiquetados

Autor(es)	Año	Idioma	Aspectos a resaltar de la revisión realizada
Francis y Kucera [108]	1979	Inglés	• Construir un corpus lingüístico no es una tarea fácil, que toma tiempo y es costosa dado lo dispendioso de la tarea. • No resuelven problemas para lenguas no tradicionales que poseen poca descripción, información y los datos de los corpus son pequeños • Para lenguas tradicionales (europeas e inglés americano) ya existen corpus muy utilizados con excelentes características como: tamaño, variedad, han sido varias veces revisados, es decir, son confiables. • Para el caso de lenguajes pobres de recursos lingüísticos, la realización de estos corpus es aún más costosa porque en la mayoría de los casos hay que hacer el etiquetado manual, dado que no hay ninguno o son escasos los corpus (o textos en la lengua) • De la revisión realizada se puede obtener una imagen del contexto metodológico para la creación de un corpus etiquetado.
Marcus et al [109]	1993	Inglés	
Kohen [111]	2005	Multilenguaje (lenguas comunidad europea)	
Ahmed y Qadir [112]	2010	Shindi	
Spoustová y Spousta [113]	2012	Checo	
Ariaratnam, et al [36]	2014	Tamil	
Sing and Banergee [114]	2014	Bhojpuri	
Scherrer, et al [115]	2014	Multilingual (alemán, francés, italiano, e inglés)	

Tabla 2.8. Análisis comparativo del estado del arte revisado sobre corpus lingüísticos etiquetados

Conjuntos de etiquetas

Autor(es)	Año	Idioma	Aspectos a resaltar de la revisión realizada
Baskaran, et al [91]	2008	India	• Forma de uso y descripción de los corpus existentes. • No hay recomendaciones sobre la unificación de etiquetas para lenguas diferentes a las tradicionales • Conjunto de etiquetas en la mayoría de las ocasiones es dependiente del lenguaje, pero existen iniciativas en pro de la Unificación de etiquetas para simplificar el proceso de etiquetado que se pueden aplicar a las lenguas
Petrov, et al [10]	2012	Multilenguaje (25 lenguas)	
Dinakaramani, et al [110]	2014	Indonesia	

Tabla 2.9. Análisis comparativo del estado del arte revisado sobre conjuntos de etiquetas para los corpus

Algunas brechas encontradas en la revisión sobre corpus son:

- Es difícil encontrar recursos lingüísticos para lenguas no tradicionales, adicionalmente, no existe una metdología o conjunto de pasos claramente definidos para la construcción de corpus etiquetados para este tipo de lenguas.

- No existe un corpus etiquetado para el nasa yuwe, la cual es una lengua que se encuentra en proceso de descripción, para hacer su procesamiento con etiquetadores, se requiere construir uno propio.

2.4 Síntesis

Del análisis realizado, se puede resaltar que:

- A pesar de las diferentes propuestas, todavía es necesario buscar soluciones para la construcción de POST menos complejos que igualen o mejoren los resultados reportados en la actualidad y que además sean eficientes en el uso de recursos computacionales.

- Son pocas las metaheurísticas utilizadas hasta el momento para modelar el problema de etiquetado, y los resultados obtenidos con su uso son prometedores. Por tanto, es posible pensar en utilizar otras metaheurísticas para este problema, especialmente si se tiene en cuenta lo postulado por los teoremas Non Free Lunch de optimización.

- Existen muchos corpus reconocidos y ampliamente utilizados en el problema de etiquetado para lenguas tradicionales, como el caso del inglés, pero para lenguas no tradicionales no son fáciles de encontrar.

- No existen procesos metodológicos claramente definidos para la creación de corpus etiquetados, y como se pudo apreciar en su mayoría el etiquetado inicial se ha realizado manualmente, lo cual convierte esta tarea en costosa y dispendiosa.

- En las diferentes propuestas se evidencia la falta de claridad sobre cómo se pueden replicar los experimentos y cómo se han realizado las comparaciones con las propuestas previas de cada artículo.

- Antes de la realización de esa tesis no existía un corpus lingüístico etiquetado para la lengua nasa yuwe, la cual es una lengua independiente, es decir, que no se parece a otra, por tanto, no se contó con un referente de una lengua o trabajo similar para su construcción.

Capítulo 3

Corpus lingüísticos etiquetados

En este capitulo, se presentan en primera instancia una breve descripción del proceso de construcción del corpus etiquetado para nasa yuwe, incluyendo detalles metodológicos y resultados del corpus construido. En segunda instancia, se presenta una breve descripción del corpus Brown para el idioma inglés utilizado utilizado en esta tesis para los experimentos presentados en el capítulo 5. Finalmente, se presenta una síntesis de los resultados mostrados en esta sección.

3.1 Construcción del corpus lingüístico etiquetado para nasa yuwe

3.1.1 Descripción del asesor

Para la construcción del corpus etiquetado de nasa yuwe se contó con la asesoría del profesor Tulio Rojas Curieux, Doctor en lingüística de la Universite Paris VII, docente del departamento de antropología de la Universidad. El profesor Tulio Rojas lleva más de 30 años investigando temas relacionados con la vida de los pueblos indígenas, su proceso de organización y sus lenguas. Es el director del Grupo de Estudios Lingüísticos, Pedagógicos y Socioculturales del suroccidente colombiano - GELPS, cuyo objetivo central es avanzar en la investigación y conocimiento de la realidad lingüística, pedagógica y cultural del suroccidente colombiano para la construcción de una educación acorde con la diversidad del país. Su amplia trayectoria como docente, investigador asociado y director de proyectos de investigación en etnolingüística se puede apreciar en el Sistema de Información de Colciencias - CVLAC[9].

3.1.2 Proceso metodológico del diseño y construcción del corpus etiquetado para nasa yuwe

Para el diseño y construcción del corpus lingüístico para la lengua nasa yuwe, presentado en esta sección, se aplicó el Patrón de Investigación Iterativo (Iterative Research Pattern [17]), el cual se presenta en la primera parte de esta sección,

[9] http://scienti.colciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0000061425

seguidamente se describen las características principales del corpus y finalmente, se muestra la alineación del corpus construido para la lengua nasa yuwe con el conjunto de etiquetado Universal [10]. Proceso seguido para la construcción de corpus lingüístico etiquetado para la lengua nasa yuwe

El proceso seguido para el etiquetado del corpus fue manual desarrollado en dos iteraciones, así:

- En la primera Iteración, se etiquetan las palabras de cada oración, originando la primera versión del corpus y posteriormente se hicieron los ajustes al corpus mediante un ejercicio que amplió el número de expertos y se usó la técnica Delphi para el juicio de expertos, generando así la segunda versión del corpus etiquetado para la lengua nasa yuwe.

- En la segunda iteración, se etiquetan manualmente las palabras teniendo en cuenta el conjunto de etiquetado universal, tomando como base la segunda versión del corpus nasa etiquetado, obteniendo así la tercera versión del corpus etiquetado nasa y posteriormente, se desarrolla un ejercicio similar al desarrollado en la primera versión (Técnica Delphi), para finalmente conseguir la versión 4 o final del corpus.

- La corrección y ajustes a las versiones del corpus tanto en la primera como en la segunda iteración fue realizada de forma manual.

- La curva de aprendizaje para la tarea de etiquetado manual fue alta, debido a que era la primera vez que la lengua nasa era sometida a esta tarea, es decir, los profesores hablantes de nasa yuwe, quienes enseñan esta lengua en las instituciones educativas de sus resguardos, no habían entrado al detalle de estudiar el rol de una palabra en una oración, para el caso de esta lengua, por tanto, se requirieron varias sesiones para la comprensión de lo que se deseaba obtener en esta tarea y para acordar el proceso que se deseaba seguir. Para la tarea de etiquetado se trabajó en sesiones de 6 horas semanales durante un periodo aproximado de 6 meses, es decir, que la velocidad de etiquetado fue muy baja al principio, lo cual fue mejorando con el tiempo.

- La forma que se utilizó para hacer el etiquetado del corpus nasa fue similar a la que se usó con el Corpus Brown (uno de los más usados a nivel mundial) [108], es decir, en cada oración, cada palabra fue etiquetada con su correspondiente rol, así como se muestra en la **Figura 3.1**, esto con el fin de facilitar su posterior procesamiento.

The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent/jj primary/nn election/nn produced/vbd ``/`` no/at evidence/nn "/" that/cs any/dti irregularities/nns took/vbd place/nn ./.

Figura 3.1. Ejemplo oración etiquetada en Corpus Brown

En la **Figura 3.2**, se puede apreciar un resumen del proceso seguido para obtener el corpus lingüístico etiquetado para la lengua nasa yuwe y la alineación de este con el etiquetado Universal, siguiendo las fases propuestas por Kevin Pratt [17] y utilizando dos iteraciones, como se mencionó anteriormente.

Figura 3.2. Resumen del proceso de etiquetado del corpus para nasa yuwe

A continuación, se presenta una breve descripción del proceso llevado a cabo para la construcción del corpus etiquetado para nasa yuwe, mostrado en la **Figura 3.2**.

Breve descripción de los pasos seguidos en la iteración 1.

En la **Tabla 3.1** a **Tabla 3.4** se describen los pasos realizados en cada una de las fases mostradas en la **Figura 3.2**.

Fase	Paso	Descripción
Observación	1	Revisión del estado del arte, enfocándose en los trabajos relacionados con la construcción de corpus, que involucraban: • Conjunto de etiquetados y el proceso de definición de nuevos conjuntos de etiquetados • Corpus lingüísticos de etiquetado existentes y su correspondiente evaluación de los conjuntos de etiquetado. • Métodos de etiquetado manual (etiquetadores o expertos en lingüística), combinaciones de manual y automático o a través de comparaciones con otros corpus existentes para el lenguaje. • Diferentes fuentes de datos (colecciones de la web o transcripciones de conversaciones, entrevistas, diccionarios o corpus existentes)
	2	Se realizó una revisión de posibles categorías de etiquetado para la lengua nasa yuwe, usando la bibliografía existente, y finalmente, se decidió utilizar la propuesta del profesor Tulio Rojas [75]. Cabe resaltar que en el desarrollo de esta investigación se contó con el apoyo y asesoría del profesor Tulio Rojas.

Tabla 3.1. Descripción de los pasos de la fase observación de la iteración 1

Fase	Paso	Descripción
Identificación	3	Se revisaron las categorías de las palabras nasa, presentadas en la sección 2.15 (ver página 17) en conjunto con los profesores hablantes de nasa yuwe y el lingüista experto. Las categorías (predicativa, cualificativa, nombre, deíctico conector, pronombre y puntuación) fueron revisadas, pero se resalta que se deja abierta la posibilidad de adicionar otras categorías, si se presentara alguna situación que lo ameritara. Lo anterior teniendo en cuenta que es la primera vez que se realiza un trabajo de esta naturaleza para este idioma.
	4	Se capacitó al profesor hablante de nasa yuwe que participó en el proceso de etiquetado, con el fin de instruirlo sobre cómo realizar la tarea de etiquetado de cada una de las palabras en las oraciones.
	5	Se seleccionaron los textos escritos en nasa yuwe de la colección de prueba [9] construida para esta lengua. Dada la naturaleza experimental de esta tarea, solo 8 textos fueron incluidos.
	6	Se conformó el equipo de etiquetado del corpus nasa contando con el profesor hablante de nasa yuwe, el lingüista y la estudiante de doctorado de la presente tesis y los directores de la tesis doctoral (expertos en POST).

Tabla 3.2. Descripción de los pasos de la fase identificación de la iteración 1

Fase	Paso	Descripción
Desarrollo de la solución	7	Identificación de las oraciones en cada uno de los textos seleccionados.
	8	Traducción de cada palabra nasa de los textos a castellano, con el fin de facilitar la tarea de etiquetado y la posible alineación entre los textos en nasa yuwe y su respectiva frase en castellano.
	9	Etiquetado de cada palabra en las oraciones identificadas, mediante el uso de la Técnica Delphi (para el juicio de expertos), de forma que el profesor hablante de nasa y la estudiante de doctorado después de realizar el etiquetado y donde no había consenso, se tomó el juicio del lingüista como criterio de desempate.
	10	Revisión del etiquetado de cada palabra en las oraciones nasa, para lo cual, se contó con la activa participación del lingüista.
	11	Se obtuvo la primera versión del corpus etiquetado nasa, junto con su correspondiente sistematización en una base de datos

Tabla 3.3. Descripción de los pasos de la fase desarrollo de la solución de la iteración 1

Fase	Paso	Descripción
Prueba de la solución	12	Se realizó un ejercicio con otros profesores hablantes de nasa yuwe y estudiantes de doctorado en lingüística (con amplios conocimientos en la lengua nasa) de la Universidad del Cauca, en el cual se seleccionaron 20 oraciones para que ellos realizaron el etiquetado.
	13	Se realizaron varias sesiones para discutir los resultados, en varios casos fue necesario realizar algunos ajustes al corpus etiquetado, obteniendo así la segunda versión del corpus y su debida actualización en la base de datos.

Tabla 3.4. Descripción de los pasos de la fase prueba de la solución de la iteración 1

Breve descripción de los pasos seguidos en la iteración 2.

En la **Tabla 3.5** a **Tabla 3.7**, se describen los pasos realizados en cada una de las fases de la iteración 2, mostradas en la **Figura 3.2**.

Fase	Paso	Descripción
Identificación	3	Se revisaron las categorías del etiquetado universal, en pro de hacer la futura alineación del corpus construido, a este conjunto de etiquetado.
	4	Se capacitó al profesor hablante de nasa yuwe sobre cómo hacer la alineación del corpus con relación al conjunto de etiquetado universal.
	5	Se seleccionaron los textos escritos en nasa yuwe de la colección de prueba [9] construida para esta lengua. Dada la naturaleza experimental de esta tarea, solo 8 textos fueron incluidos.

Tabla 3.5. Descripción de los pasos de la fase de identificación de la iteración 2

Fase	Paso	Descripción
Desarrollo de la solución	9	Etiquetado de cada palabra en las oraciones identificadas, mediante el uso de la Técnica Delphi (para el juicio de expertos), de forma que el profesor hablante de nasa y la estudiante de doctorado después de realizar el etiquetado y donde no había consenso, se tomó el juicio del lingüista como criterio de desempate.
	9	De la misma forma se realizó la alineación de categorías del etiquetado universal con las categorías nasa asignadas a cada una de las palabras de las oraciones.
	10	Revisión del etiquetado de cada palabra en las oraciones nasa, para lo cual, se contó con la activa participación del lingüista.
	11	Se obtuvo la tercera versión del corpus etiquetado nasa, alineado con el conjunto de etiquetado universal y su correspondiente sistematización en una base de datos.

Tabla 3.6. Descripción de los pasos de la fase de desarrollo de la solución de la iteración 2

Fase	Paso	Descripción
Prueba de la solución	12	Se efectuó un ejercicio similar al realizado en este paso en la iteración 1, para el cual se contó también con otros profesores hablantes de nasa yuwe y estudiantes de doctorado en lingüística (con amplios conocimientos en la lengua nasa) de la Universidad del Cauca, en el cual se seleccionaron 20 oraciones donde ellos realizaron el etiquetado con el conjunto de etiquetado universal [10].
	13	Se realizaron varias sesiones para discutir los resultados, en varios casos fue necesario realizar algunos ajustes al corpus etiquetado, obteniendo así la cuarta versión del corpus etiquetado nasa alineado con el conjunto de etiquetado universal [10], así como su correspondiente actualización en la base de datos.

Tabla 3.7. Descripción de los pasos de la fase prueba de la solución de la iteración 2

3.1.3 Corpus lingüístico etiquetado para nasa yuwe

Conjunto de datos.

El proceso de construcción del corpus fue todo un desafío dado que el nasa yuwe no es una lengua, de la cual se puedan encontrar recursos lingüísticos digitalizados y en el formato requeridos para su utilización en este trabajo, por tanto, se tomó como base los textos digitalizados en la colección de textos de prueba de nasa yuwe [9]. Esta colección de prueba consta de 97 documentos escritos en nasa yuwe. El contenido de los textos hace referencias a historias populares de la vida y cosmovisión nasa. También cabe resaltar que todos los textos están escritos en alfabeto unificado. Las

oraciones etiquetadas en el corpus fueron seleccionadas de 8 textos de la colección prueba de nasa yuwe [9], de tal forma que el corpus etiquetado construído quedó conformado de la manera en que se presenta en la **Tabla 3.8**.

Titulo texto nasa	Textos en Castellano	# oraciones	# palabras
Nasa vxanxi's pta'sxnxi	El origen del hombre	12	119
kutxh wala ũpxhnxi yuwe	El origen del máiz	28	345
Jũth upxhnxi yuwe	Historia de la patata	14	141
Eçxthẽ' vxuu naamu'	Historia del diablo	11	124
Ũ' tasx tuthenxi	Origen de la comida	16	225
Yu' vxaanxi yuwe	Origen del agua	40	251
Wejxa yuwe	Origen del viento	35	477
Kus	La noche	19	219
	Total	175	1091

Tabla 3.8. Descripción de los textos nasa yuwe

Conjunto de etiquetado (tagset) para nasa yuwe.

El conjunto de etiquetado para nasa yuwe utilizado fue el propuesto por [49] que fue previamente descrito en el numeral 2.1.5, al cual se le adicionó una etiqueta para signos de puntuación y una etiqueta de pronombre que fue incluida por el profesor Tulio Rojas, al momento de hacer las revisiones del corpus etiquetado, en la Tabla 3.9, se pueden apreciar las frecuencias de cada etiqueta en el corpus de nasa yuwe y en la Figura 3.3, se aprecia la distribución de las etiquetas en el corpus, teniendo una alta presencia de palabras predicativas y nombres.

Conjunto de etiquetado de Nasa yuwe	Frequencias	Probabilidades
Predicativa	657	33%
Cualificativa	230	11%
Nombre	637	32%
Conector	207	10%
Deíctico	76	4%
Pronombre	19	1%
Punctuación	176	9%

Tabla 3.9. Tagset para nasa yuwe

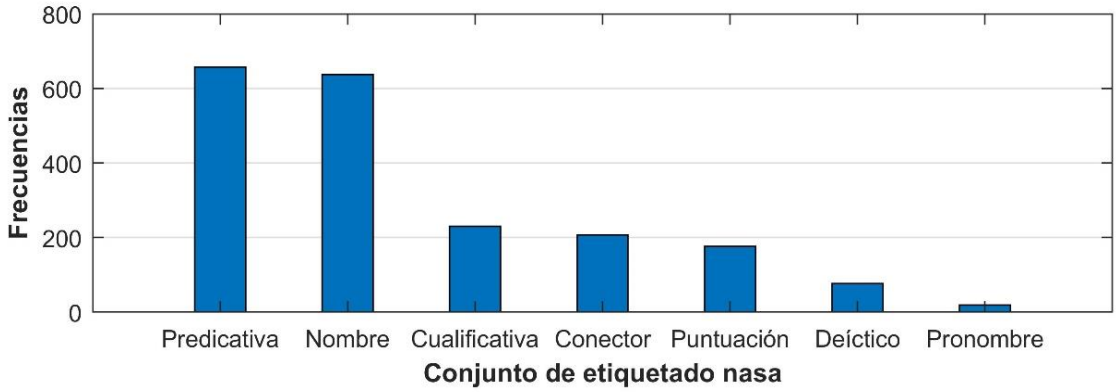

Figura 3.3. Distribución del conjunto de etiquetas en el corpus de nasa yuwe

Corpus etiquetado para nasa yuwe.

El corpus etiquetado para nasa yuwe se encuentra conformado así:

1. Palabras y Tamaño: El corpus nasa cuenta con 1176 palabras, con una longitud máxima de 19 caracteres y mínima de 1 carácter, con un promedio 7.5 caracteres. En la **Tabla 3.10**, se presentan la lista de palabras más frecuentes (top ten) en el corpus, con su respectiva etiqueta.

Posición	Palabra	Frecuencia	Posición	Palabra	Frecuencia
1	txãa	36	6	nawã	17
2	wala	27	7	aça’	17
3	txã’w	24	8	mẽh	15
4	sa’	23	9	aççxa	15
5	teeçx	19	10	u’pu’	13

Tabla 3.10. Palabras más frecuentes

2. Frases etiquetadas: Al finalizar el proceso de etiquetado manual, se obtuvieron 175 frases etiquetadas, con longitud máxima de 34 palabras y mínima de 1.

3. Ejemplo de una oración etiquetada: A continuación, en la **Figura 3.4** se presenta un ejemplo de una frase etiquetada:

txaniteya’/cualificativa kiwe/nombre wala/cualificativa u’sene’yũ’ /predicativa aça’/conector khã’sx /nombre ũskiweçxane’yũ’ /predicativa mẽh/cualificativa kũh/cualificativa jwed /nombre ksxa’w/nombre ũskiweyũ’ne’sa’/Predicativa ,/puntuación

Figura 3.4. Ejemplo oración etiquetada en Corpus nasa

Adicionalmente, en la **Tabla 3.11,** se presenta un ejemplo de las frases etiquetadas dentro del corpus, en donde se establece la respectiva etiqueta de cada palabra y el orden de la palabra en la oración.

# oración	Palabras Nasa	Etiqueta	Orden en la oración
8	Naa	Deíctico	1
8	seka’	Nombre	2
8	nmẽh	Cualificativa	3
8	Wala	Cualificativa	4
8	açxasayũ’ne’	Cualificativa	5
8	sa’	Conector	6
8	luuçxwe’sxyakh	Nombre	7
8	wẽt	Cualificativa	8
8	fxi’zeya’	Predicativa	9
8	ãjamene’ /ãhamene’	Cualificativa	10

Tabla 3.11. Ejemplo de frases etiquetadas

4. Alineación con el etiquetado Universal: En la **Tabla 3.12**, se presenta la alineación del conjunto de etiquetado para nasa yuwe en relación con el conjunto de etiquetado Universal [10]. El proceso de alineación del corpus nasa, se realizó manualmente, y no solo consistió en cambiar etiquetas (etiqueta nasa por etiquetado universal), sino que fue necesario volver a hacer revisiones detalladas para asignar la correspondiente etiqueta a cada palabra en las oraciones, especialmente en situaciones como:

- Algunas palabras que fueron etiquetadas como Nombre (Etiqueta Nasa) tuvieron que ser asignadas como Nombre (noun) o como Num (num) en el etiquetado Universal.

- Con las palabras etiquetadas como Cualificativa (Etiqueta Nasa), fue necesario revisarlas minuciosamente para definir cuál era la correspondiente etiqueta en el Etiquetado Universal (Adv o Adj).

Conjunto de etiquetado Universal	Conjunto de etiquetad del Nasa yuwe	Frequencia
Verb	Predicativa	659
Adj	Cualificativa	158
Adv	Cualificativa / Conector	213
Noun	Nombres	640
Num	Nombres / Cualificativa	3
Det	Deíctico	77
Pron	Pronombre / Conector	28
Conj	Conector	48
Prt	No aplica	-
Adp	No aplica	-
Punctuation	Punctuación	176
X	Otras palabras	-

Tabla 3.12. Alineación del TagSet para nasa yuwe

La **Figura 3.5** muestra la distribución del conjunto de etiquetado universal sobre el corpus etiquetado para nasa yuwe, una vez se ha realizado el proceso de alineación correspondiente.

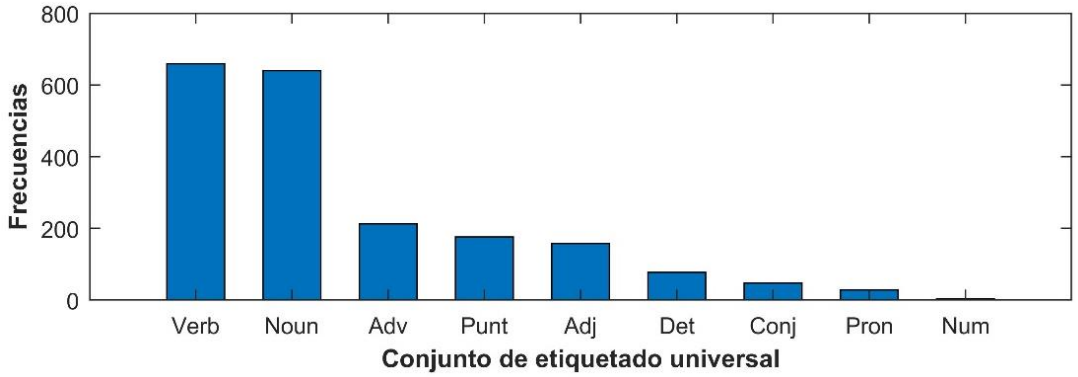

Figura 3.5. Distribución del conjunto de etiquetado universal en el corpus nasa

5. Publicación del corpus: A continuación, en la **Tabla 3.13** se presentan algunas frases seleccionadas que conforman el corpus para nasa yuwe, con el fin de permitir apreciar las características del mismo.

IDFRASE	TERMINO	ROL	ROL PETROV	ORDEN	SIGNIFICADO
4	Ne'jwe'sxyũ'	nombre	noun	1	El mayor (autoridad espiritu)
4	Puutx	cualificativa	adv	2	convivir entre ellos, juntos
4	ptamne'tayũ'	nombre	noun	3	parejas
4	.	puntuación	punct	4	punto
4	Piçthẽ'jsa'	nombre	noun	5	el hombre
4	tayne'	nombre	noun	6	TAY (NOMBRE PROPIO)
4	yaaseyu'	predicativa	verb	7	se llamaba
4	,	puntuación	punct	8	coma
4	u'ysa'	nombre	noun	9	la mujer
4	umane'	nombre	noun	10	UMA (NOMBRE PROPIO)
4	yaaseyu'	predicativa	verb	11	se llamaba
4	.	puntuación	punt	12	punto
17	kxuyyu'	cualificativa	adv	1	por allí
17	yu'a'	nombre	noun	2	el agua
17	abxya'	predicativa	verb	3	apareció
17	takhe'ne'	predicativa	verb	4	empezó
17	luuçxkna'sa	nombre	noun	5	señorita
17	u'y	nombre	noun	6	vió
17	luuçxkwesayu'	nombre	noun	7	y la niñita
17	utxane'	predicativa	verb	8	se arrimó
17	thegya'	predicativa	verb	9	a ver
17	u'j	predicativa	verb	10	ir
17	txãa	deíctico	det	11	ese
17	thẽ'sa'	nombre	noun	12	el mayor
17	dehna'wne'	predicativa	verb	13	como si estuviera dormido
17	thegu'	predicativa	verb	14	aparecía
17	aççxa	conector	conj	15	luego entonces
17	,	puntuación	punct	16	coma
121	txãa	deíctico	det	1	ese
121	wasakwe	nombre	noun	2	señorita
121	amehte'naw	cualificativa	adj	3	despacio
121	ne'we'w	predicativa	verb	4	hablo
121	:	puntuación	punct	5	dos puntos
121	Ũ'kwe'	nombre	noun	6	no tiene significado
121	yu'	nombre	noun	7	no tiene significado

Tabla 3.13. Algunas frases del corpus del corpus etiquetado nasa yuwe. Parte I

IDFRASE	TERMINO	ROL	ROL PETROV	ORDEN	SIGNIFICADO
121	nxji'tsa'	predicativa	verb	8	digo
121	i'kwe'sx	pronombre	pronoun	9	ustedes
121	jxukasatx	nombre	noun	10	todos
121	yu'	nombre	noun	11	el agua
121	nvxĩhtna	predicativa	verb	12	lo dejo
121	ũsu'	predicativa	verb	13	estuvo
121	pakaçxhyuhpa	cualificativa	adv	14	hasta siempre
121	peejxme	nombre	noun	15	sin necesidad
121	sa	conector	adv	16	entonces
121	kiwe's	nombre	noun	17	la tierra
121	ĩkh	nombre	noun	18	el lago
121	dxikhe	nombre	noun	19	hacia el fondo
121	akhme'	predicativa	verb	20	no lo hecho
121	.	puntuación	punct	21	punto

Tabla 3.14. Algunas frases del corpus del corpus etiquetado nasa yuwe. Parte II

Como se puede apreciar en la **Tabla 3.13** y **Tabla 3.14**, la traducción literal al castellano que se realizó de cada una de las palabras de las oraciones no es muy significativa, como si lo es teniendo en cuenta todo el contexto de la oración. Esta traducción se hizo para facilitar el proceso de etiquetado.

A continuación, en la **Tabla 3.15**, se presentan las mismas oraciones del corpus con el conjunto de etiquetado nasa y el conjunto de etiquetado universal.

IDFRASE	Corpus etiquetado nasa yuwe	Corpus nasa etiquetado Universal
4	ne'jwe'sxyũ'/nombre puutx/cualificativa ptamne'tayũ'/nombre ./puntuación piçthẽ'jsa'/nombre tayne'/nombre yaaseyu'/predicativa ,/puntuación u'ysa'/nombre umane'/nombre yaaseyu'/predicativa ./puntuación	ne'jwe'sxyũ'/noun puutx/adv ptamne'tayũ'/noun ./punct piçthẽ'jsa'/noun tayne'/noun yaaseyu'/verb ,/punct u'ysa'/noun umane'/noun yaaseyu'/verb ./punt
17	kxuyyu'/cualificativa yu'a'/nombre abxya'/predicativa takhe'ne'/predicativa luuçxkna'sa/nombre u'y/nombre luuçxkwesayu'/nombre utxane'/predicativa thegya'/predicativa u'j/predicativa txãa/deíctico thẽ'sa'/nombre	kxuyyu'/adv yu'a'/noun abxya'/verb takhe'ne'/verb luuçxkna'sa/noun u'y/noun luuçxkwesayu'/noun utxane'/verb thegya'/verb u'j/verb txãa/Det thẽ'sa'/noun dehna'wne'/verb thegu'/verb aççxa/conj ,/punct

IDFRASE	Corpus etiquetado nasa yuwe	Corpus nasa etiquetado Universal
	dehna'wne'/predicativa thegu'/predicativa aççxa/conector ,/puntuación	
121	txãa/deíctico wasakwe/nombre amehte'naw/cualificativa ne'we'w/predicativa :/puntuación ũ'kwe'/nombre yu'/nombre nxji'tsa'/predicativa i'kwe'sx/pronombre jxukasatx/nombre yu'/nombre nvxĩhtna/predicativa ũsu'/predicativa pakaçxhyuhpa/cualificativa peejxme/nombre sa/conector kiwe's/nombre ĩkh/nombre dxikhe/nombre akhme'/predicativa ./puntuación	txãa/Det wasakwe/noun amehte'naw/adj ne'we'w/verb :/punct ũ'kwe'/noun yu'/noun nxji'tsa'/verb i'kwe'sx/pronoun jxukasatx/noun yu'/noun nvxĩhtna/verb ũsu'/verb pakaçxhyuhpa/adv peejxme/noun sa/adv kiwe's/noun ĩkh/noun dxikhe/noun akhme'/verb ./punct

Tabla 3.15. Ejemplo corpus del corpus etiquetado nasa yuwe conjunto de etiquetado nasa y universal

El corpus etiquetado para nasa yuwe se encuentra publicado en este link[10]. Tambien en los anexos digitales 1 a 3 (Anexo Digital 1 CorpusNasa Etiquetado Nasa, Anexo Digital 2 Corpus nasa Etiquetado Universal, Anexo Digital 3 CorpusNasa Etiquetado Completo), se encuentra el corpus nasa yuwe completo.

3.2 Breve descripción del Corpus Brown (inglés)

3.2.1 Descripción del corpus etiquetado

El corpus Brown [108] es una recopilación de textos en inglés realizada por Henry Kucera y Nelson Francis, está conformado por 500 archivos de texto plano, agrupados en las categorías presentadas en la **Tabla 3.16**

[10] https://drive.google.com/open?id=0BzjrO-_P-eYTekRaclQ2UFc0aTA

Tipo de documento	Cantidad de ejemplos
Prosa Informativa	**374 ejemplos**
A. Press: reportage	44
B. Press: editorial	27
C. Press: reviews	17
D. Religion	17
E. Skill and hobbies	36
F. Popular lore	48
G. Belles-lettres	75
H. Miscellaneous: government & house organs	30
J. Learned	80
Prosa imaginative	**126 ejemplos**
K: fiction: general	29
L: fiction: mystery	24
M: fiction: science	6
N: fiction: adventure	29
P. Fiction: romance	29
R. Humor	9

Tabla 3.16. Descripción de texto Corpus Brown**. Fuente:** *[108]*

El conjunto de etiquetas de este corpus esta conformado por 87 categorias, las cuales se subdividen, obteniendo así un total de 472 etiquetas. La **Tabla 3.17,** presenta algunos ejemplos del conjunto de etiquetado Brown general. En el Anexo B, se encuentra el detalle completo de este conjunto de etiquetas.

Tag	Description	Examples	Tag	Description	Examples
.	sentence closer	. ; ? !	AP	post-determiner	many, several,next
(	left paren		AT	article	a, the, no
)	right paren		BE	Be	
*	not, n't		BED	were	
--	Dash		BEDZ	Was	
,	Comma		BEG	being	
:	Colon		BEM	Am	
ABL	pre-qualifier	quite, rather	BEN	been	
ABN	pre-quantifier	half, all	BER	are, art	
ABX	pre-quantifier	Both	BEZ	is	

Tabla 3.17..Descripción de algunas etiquetas Brown. **Fuente:** *[108]*

Para el caso de esta tesis, el corpus Brown [108] fue seleccionado para el desarrollo de los experimentos con el inglés, debido a que es ampliamente utilizado en los diferentres trabajos de etiquetado revisados.

3.3 Síntesis

Los resultados presentados en esta sección son principalmente:

- La formalización del proceso de construcción de un corpus etiquetado para una lengua como es el nasa yuwe, el cual se presenta con cierto nivel de detalle constituyéndose en un referente para futuros trabajos sobre esta lengua u otra, sobre la que se desee realizar un trabajo similar.

- El corpus etiquetado para nasa yuwe, a pesar de su tamaño, establece un aporte significativo para continuar trabajos sobre esta lengua, que permitan favorecer el desarrollo de herramientas computacionales favoreciendo procesos de enseñanza – aprendizaje y visibilización a través de recursos informáticos. Lo cual puede ser obtenido a través de las diferentes aplicaciones que tiene el procesamiento de lenguaje natural.

Finalmente, cabe resaltar que los resultados obtenidos se consiguieron a través de una revisión exhaustiva de trabajos que involucran el uso y la construcción de corpus etiquetados y la definición o uso de conjuntos de etiquetas tanto para lenguas tradicionales como no tradicionales, lo cual permitió:

- Sistematizar y definir el proceso a seguir para la construcción del corpus etiquetado para la lengua nasa yuwe.

- Conocer los diferentes corpus etiquetados utilizados para el idioma inglés y seleccionar el corpus Brown para realizar los experimentos de esta tesis.

Capítulo 4

Algoritmo Memético para la identificación de partes del discurso

El proceso metodológico utilizado para desarrollar el algoritmo memético de etiquetado propuesto en esta sección, fue el Patrón de Investigación Iterativo (Iterative Research Pattern [17]). En este capitulo se presenta en la sección 4.1, el problema de etiquetado desde un enfoque estadístico, en la sección 4.2, se aborda el problema de etiquetado como un problema de optimización, en la sección 4.3, se presenta el algoritmo memético propuesto, en la sección 4.4 se presentan algunas consideraciones sobre el proceso seguido para la construcción del algoritmo y su implementación. Finalmente, se presenta una síntesis de los resultados mostrados en este capítulo.

4.1 El problema de Identificación de partes del discurso

Como se mencionó en la sección 2, para determinar la secuencia de etiquetas correcta, se tienen en cuenta la relación entre la morfología y la sintaxis dentro de la oración (el contexto), para lo cual un etiquetador codifica y usa las restricciones impuestas por estas relaciones, es decir, limita el contexto a unas pocas palabras alrededor de la palabra a etiquetar y haciendo uso de la información provista por este contexto e ignorando el resto [16].

Para llevar a cabo esta tarea, un algoritmo para POST realiza de forma resumida lo siguiente [99]:

Asigna la mejor secuencia de etiquetas para una oración $S = \{w_1, w_2, .., w_n\}$, donde w_i indica la i-ésima palabra a ser etiquetada, lo cual se expresa formalmente como la secuencia de etiquetado $T*$, que es la solución óptima con respecto a todas las otras posibles soluciones $\mathrm{T} = \{\mathrm{T}^1, \mathrm{T}^2, \dots, \mathrm{T}^{N(n)}\}$, donde cada $\mathrm{T}^i = \{t_1^i, t_2^i, \dots, t_n^i\}$, representa una etiqueta candidata, $N(n) = \prod_{i=1}^{n} k_i$, es el número de todas las posibles secuencias de etiquetas para el etiquetado de una oración, donde ki es el número valido de etiquetas para la palabra i-ésima en la oración (por ejemplo w_i) [99].

La interpretación Bayesiana de esta tarea considera todas las posibles secuencias de etiquetas [29] y busca encontrar la secuencia de etiquetado más probable para la oración a etiquetar. Desde el punto de vista estadístico [28], un etiquetador es definido como la tarea de escoger una secuencia de etiquetas con la máxima probabilidad. Sin embargo, en métodos estadísticos (Modelos Ocultos de Markov, HMM), se hacen dos supuestos [125]:

- La probabilidad de que una palabra aparezca es dependiente sólo de su propia etiqueta, la cual es independiente de las otras palabras alrededor de ella, y de las otras etiquetas alrededor de ella.

- La etiqueta de una palabra sólo depende de K etiquetas anteriores (probabilidad de transición).

Por tanto, simplificando con Bayes se busca una solución en T* [99] que satisfaga la **Ecuación 4.1**

$$T^* = \arg\max \mathrm{P}(T|S) = \arg\max_{T \in \mathfrak{T}} \left[\prod_{i=0}^{n-1} \mathrm{P}(w_i|t_i)\, \mathrm{P}(t_i|t_{i-1}) \right]$$

Ecuación 4.1. Problema de etiquetado. **Fuente:** [99]

donde T^* es la mejor secuencia de etiquetas, $\mathrm{P}(w_i|t_i)$ representa la probabilidad de una palabra en una oración dado una etiqueta y $\mathrm{P}(t_i|t_{i-1})$ representa la probabilidad de una etiqueta dada la etiqueta previa [99].

4.2 Identificación de partes del discurso como un problema de optimización

Tomando como referente trabajos previos [19, 98, 99], se han tenido en cuenta las siguientes consideraciones:

1. Para representar el etiquetado como un problema de optimización es necesario tener en cuenta que el objetivo consiste en encontrar las mejores etiquetas para una secuencia de palabras.

2. Las posibles etiquetas de una palabra son identificadas acorde con el conjunto de etiquetado propuesto por Petrov y otros [10].

3. Las soluciones candidatas son la secuencia más probable de etiquetas para un conjunto de palabras a etiquetar, las cuales son obtenidas de las diferentes etiquetas que una palabra puede tomar teniendo en cuenta su contexto.

4. El contexto seleccionado para calcular la etiqueta más probable de una palabra considera una ventana de tamaño tres: la palabra predecesora, la palabra a etiquetar y la palabra sucesora.

5. La función objetivo, se calcula como se muestra en la **Ecuación 4.2**.

$$Fitness = \prod_{i=0}^{n-1} P(w_i|t_j)\ P(t_i\,|t_{i-1}, t_{i+1})$$

Ecuación 4.2. POST como problema de optimización con contexto (Trigrama). **Fuente:** [19]

4.3 Algoritmo memético de etiquetado propuesto

Siguiendo lo comentado en la anterior sección, la propuesta involucra los siguientes aspectos:

1. Se ha seleccionado el algoritmo metaheurístico Global-Best Harmony Search como base de esta propuesta, teniendo en cuenta que se desempeña mejor que la versión original de Harmony Search [19, 98].

2. Para efectos de este trabajo, cada solución se representa como se muestra en la **Figura 4.1**.

- En esta figura, se cuenta con un vector del tamaño del número de palabras en una oración (cada palabra se representa con una posición del vector), que contiene las etiquetas asignadas a cada palabra, desde la posición 0 a la posición $n-1$ $(\mathrm{T}_0, \mathrm{T}_1, \dots, \mathrm{T}_{\mathrm{n}-1})$.

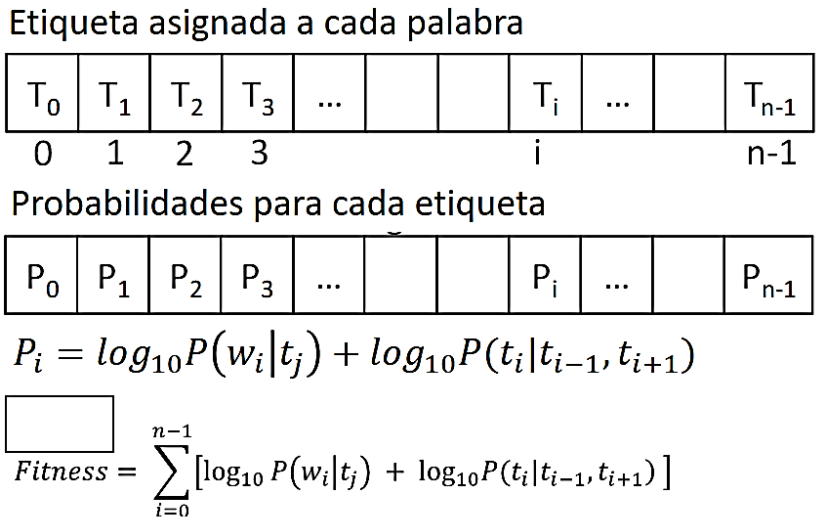

Figura 4.1. Representación de la solución

Se utiliza otro vector que contiene la probabilidad acumulada de cada palabra etiquetada, y su relación con su predecesor y su sucesor, calculada como $P_i = log_{10}P(w_i|t_j) + log_{10}P(t_i|t_{i-1}, t_{i+1})$. Como se puede apreciar en la **Figura 4.1**, el contexto seleccionado para la palabra a etiquetar es un trigrama (predecesora, palabra a etiquetar, sucesora).

- Finalmente, se tiene otro campo que es para almacenar el valor de la función fitness, la cual incluye la probabilidad $P(w_i|t_j)$, de cada una de las posibles etiquetas (t_j) de la palabra (w_i) para etiquetar, la cual es independiente de su contexto, que corresponde a la probabilidad de las etiquetas de la palabra predecesora y de la sucesora de la palabra a etiquetar, dada la etiqueta de la palabra.

- Para evitar que la evaluación de la función fitness se convierta en cero después de varias multiplicaciones de las correspondientes probabilidades (valores que están entre cero y uno) (ver **Ecuación 4.2**), la ecuación se transformo aplicando la suma de logaritmos a cada uno de los productos de la función objetivo, como se aprecia en la **Ecuación 4.3**. Adicionalmente, fue asignado un valor por defecto cuando se evalúa un trigrama que no está en los datos de entrenamiento, se utilizó un valor cercano a cero, definido como 0.000001.

$$Fitness = \sum_{i=1}^{n} \left[\log_{10} P\left(w_i \middle| t_j\right) + \log_{10} P\left(t_i \middle| t_{i-1}, t_{i+1}\right) \right]$$

Ecuación 4.3. Evaluación de la Función Fitness. **Fuente:** Adaptado de *[19]*

3. El algoritmo de etiquetado GBHS propuesto (GBHS tagger), se presenta a continuación, (ver **Algoritmo 4.1**). Este algoritmo muestra un resumen de la estructura del algoritmo memético para etiquetado basado en GBHS que incluye optimización local para la mejor armonía en la memoria armónica (HM).

Algoritmo 4.1 GBHS para etiquetado propuesto

1. Definir parametros: HMS, NI, HCMR, PARMin, ParMax, ProbOpt, Alpha, MaxNeighbors
2. Inicialización aleatoria de HM o inicialización mejorada usando el parámetro Alpha
3. **para** $i = 1$ to NI **hacer**
4. $PAR \leftarrow PARMin + (PARMax - PARMin) \times \left({}^{i}/_{NI}\right)$ /* definición de PAR */
5. **para** $j = 1$ to n **hacer** /* para cada palabra en la oración */
6. **si** (Activo[j] == true) **entonces** /* ¿la palabra actual tiene más de una posible etiqueta? */
7. **si** $(U(0,1) \leq HMCR)$ **entonces** /* memory consideration */
8. $x_j' \leftarrow x_j^p$, donde $p \sim U(1, \ldots, HMS)$
9. **Si** $(U(0,1) \leq PAR$ **entonces**
10. $x_j' \leftarrow x_k^{best}$, donde best es el índice de la mejor armonía en HM and $k \sim U(1,n)$
11. **fin_si**
12. **sino** /* selección aleatoria */
13. $x_j' \leftarrow LB_j + r \times (UB_j - LB_j)$
14. **fin_si**
15. **sino**
16. x_j' ← UnicaEtiquetaParaLaPalabra(j)
17. **fin_si**
18. **fin_para**
19. **mientras** (visitado (x_j')) **hacer** /* Si la solución ha sido visitada antes */
20. **si** (Activo[j] == true) **entonces /*** mutar la nueva armonía (x_j')
21. Se selecciona aleatoriamente una diferente a la actual
22. **fin_mientras**
23. Evaluar la función fitness de la nueva armonía (x_j') a través de la **Ecuación 4.3**
24. **si** (fitness (x_j') > fitness de la peor armonía en HS) **entonces**
25. HM[pos_peor] ← x_j' /* reemplazar la peor en la memoria armónica */
26. **fin_si**
27. **si** $(U(0,1) < ProbOpt)$ **entonces**
28. Aplicar Optimización Local a la mejor armonía en HM
29. **fin_si**
30. **fin_para**
31. **retornar** la mejor armonía en HM

Este algoritmo presenta las siguientes características:

- Usa los mismos parámetros de su versión original (GBHS), como son: HMCR, PARMin, PARMax, HMS, y NI (ver línea 1 del algoritmo propuesto en **Algoritmo 4.1**).

- Adicionalmente, se han definido tres parámetros a saber:
    - La probabilidad de optimización (ProbOpt), que controla el porcentaje de veces que se realiza el proceso de optimización local, este proceso se aplica a la mejor armonía (solución) en la memoria armónica.

    - El número de vecinos (MaxNeighbors) que se evalua en el proceso de optimización local.

    - El parámetro Alpha que controla si los componentes de cada armonía en la población son generados aleatoriamente de sus posibles etiquetas o tomados de la etiqueta con mayor probabilidad.

- Como se aprecia en la línea 2 del **Algoritmo 4.1**, se debe realizar la inicialización de la memoria armónica la cual normalmente se hace aleatoriamente, como se puede apreciar en el **Algoritmo 4.2**.

Algoritmo 4.2. Inicialización aleatoria de la memoria armónica

1. **Para cada** solución en la memoria armónica **hacer**
2. Se llena el vector de etiquetas para cada palabra de la oración, seleccionado aleatoriamente una etiqueta probable
3. Se llena el vector de probabilidades para cada etiqueta asociada a la palabra teniendo en cuenta el trigrama.
4. Se evalúa el fitness y se guarda en la solución
5. **Fin para cada**
6. Se obtiene la memoria armónica

- Para efectos de esta propuesta se ha introducido una mejora en la inicialización de la memoria armónica, la cual evalúa el párametro Alpha ($U(0,1) \leq Alpha$)**,** para crear el vector de etiquetas de cada palabra en la oración con la etiqueta más probable, de lo contrario utiliza la inicialización aleatoria normal, como se puede apreciar en el **Algoritmo 4.3**.

Algoritmo 4.3. Inicialización mejorada de la memoria armónica

1. **Para cada** solución en la memoria armónica **hacer**
2. **si** $(U(0,1) \leq Alpha)$ **entonces**
3. Se llena el vector de etiquetas para cada palabra de la oración, con la etiqueta más probable
   sino
4. Se llena el vector de etiquetas para cada palabra de la oración, seleccionando aleatoriamente una etiqueta probable
5. **fin_si**
6. Se llena el vector de probabilidades para cada etiqueta asociada a la palabra teniendo
7. en cuenta el trigrama
8. Se evalúa el fitness y se guarda en la solución
9. **fin para cada**
10. Se obtiene la memoria armónica

- En la línea 3 del **Algoritmo 4.1**, se inicia el proceso de etiquetado, teniendo en cuenta el número de improvisaciones (NI) a ejecutar.

- En la línea 4, se establece el parámetro PAR, teniendo en cuenta los límites definidos.

- En la línea 5, se hace el proceso de etiquetado para cada una de las palabras en la oración, así:
  - Para esta propuesta, se ha introducido, una revisión previa sobre la cantidad de etiquetas que una palabra pueda tener, es decir, si la palabra a etiquetar tiene varias posibles etiquetas o una sola (ver línea 6, **Algoritmo 4.1**).
    - Si tiene varias opciones entonces se genera el nuevo improsivo (armonía) a través de los parámetros HMCR y PAR, ó a través de la Selección Aleatoria, como se aprecia en las líneas 7 a 15.
    - Si la palabra sólo tiene una posible etiqueta, se le asigna esta etiqueta (ver líneas 16 a 17).
  - También en esta propuesta, se ha introducido una revisión a la nueva solución generada, es decir, si la nueva armonía (x'_j) es una solución ya visitada previamente, entonces, esta solución muta (ver líneas 20 a 23 del **Algoritmo 4.1**), lo cual se ejecuta hasta obtener una solución no visitada.

Esta revisión adicional, favorece que el algoritmo no se quede atrapado en óptimos locales.

- Para llevar el registro de las soluciones que ya se han visitado se utiliza una memoria tabú explícita (la solución completa).

- La mutación de la solución se hace mediante la modificación de uno de los genes de la solución, siempre y cuando exista esta opción, es decir, las palabras puedan tener otras posibles etiquetas.

- En la línea 24 del **Algoritmo 4.1**, se aprecia la evaluación de la función objetivo (fitness).

- En las líneas 25 y 26 del **Algoritmo 4.1**, se puede apreciar que, si el valor de la función objetivo de la nueva armonía es mejor que el valor de la función fitness de la peor armonía en la memoria armónica, se reemplaza esta última por la nueva armonía.

- Esta propuesta introduce en las líneas 28 a 30 del **Algoritmo 4.1**, el uso del parámetro de optimización local (ProbOpt), el cual introduce conocimiento al problema de etiquetado. El proceso de optimización se aplica o no basado en dicha probabilidad sobre la mejor armonia de la memoria armónica.

- El **Algoritmo 4.4**, presenta la estructura del optimizador local, mencionado anterioremente, (ver línea 29 del **Algoritmo 4.1**), el cual es un algoritmo adaptado de la metaheurística, Ascenso a la Colina (Hill Climbing) [3], que busca evaluar el máximo número de vecinos de la armonía. Para el caso de esta propuesta, se realiza así:
  - La definición del vecino se hace buscando modificar la asignación de etiqueta de la palabra con la peor probabilidad (P_i) en la armonía a optimizar.

  - La etiqueta asignada a la palabra seleccionada es cambiada, buscando mejorar el valor de la función objetivo.

  - Se evalúa, si la nueva armonía es mejor que la original, si es el caso se reemplaza esta última por la nueva.

- Para crear el próximo vecino, se tiene en cuenta que la palabra a seleccionar para cambiarle la etiqueta debe ser diferente de las anteriormente modificadas.

- Este proceso se repite teniendo en cuenta, la cantidad máxima de vecinos que se deben evaluar, lo cual es previamente establecido con el parámetro MaxNeighbors.

- La restricción de no repetir una palabra previamente modificada, evita sobre explotar la misma palabra en la armonía. Remover la palabra con la peor probabilidad es una mejor alternativa de optimización que la selección aleatoria (conocimiento del problema).

Algoritmo 4.4. Optimización local (armoníaActual) (Hill Climbing(current))

```
1.  ListaTabuDePalabras ← ∅
2.  para i=1 to MaxNeighbors do
3.          t ← índice de la palabra con la probabilidad mas baja Pi que no existe en la
4.                  ListaTabuDePalabras que tienen más de una posible etiqueta
5.          Si (t == ∅) entonces salir /* finaliza el proceso de optimización local */
6.          ListaTabuDePalabras.adicionar(t) /* esta palabra no se usada otra vez */
7.          nuevaArmonia ← Copiar(armoníaActual)
8.          Cambiar etiqueta (aleatoriamente) de la palabra t en la nueva armonía
9.          Si (Fitness(nuevaArmonía) > Fitness(armoníaActual) entonces
10.                 ArmoníaActual ← nuevaArmonía
11.         Fin_Si
12. Fin_para
```

- Finalmente, como se muestra en la línea 32 del **Algoritmo 4.1**, GBHS Tagger retorna la mejor armonía de la memoría armónica.

4.4 Consideraciones generales sobre el algoritmo propuesto

4.4.1 Proceso seguido para obtener el algoritmo

Como se mencionó al inicio el desarrollo de esta propuesta estuvo enmarcada en el patrón de investigación Iterativo, por tanto, la obtención del algoritmo memético de etiquetado pasó por varias iteraciones en donde en cada una se le fueron incluyeron

nuevos aspectos hasta que finalmente se obtuvo el algoritmo presentado en la sección anterior.

En primera instancia, se desarrolló el etiquetador aleatorio (Azar) el cual permitió organizar la estructura del algoritmo a proponer. Seguidamente, se implementó otro algorimto de la línea base llamado HSTagger, el cual utiliza la metaheurística Harmony Search.

En segunda instancia, se construyó un etiquetador que, inicialmente sólo involucraba la adaptación del algoritmo metaheurístico Global-Best Harmony Search al problema de etiquetado. Teniendo este etiquetador se incluyó la estrategia de optimización local, el cual involucra un parámetro que se definió como ProbOpt, que controla la probabilidad de utilizar este optimizador local en la mejor armonía de la memoría armónica. Se realizaron varias evaluaciones con conjuntos de datos pequeños, para definir cuáles serían los valores a utilizar en el proceso de optimización. De tal forma, que se obtuvo la primera versión del algoritmo memético de etiquetado, que involucra el optimizador local y la inicialización aleatoria de la memoría armónica, esta versión se denomino GBHS Tagger, a la cual se le definieron 4 valores al parámetro ProbOpt, como fueron sin optimización (0.0), con un valor de optimización de 0.3, 0.5 y 0.7, y se nombraron así: GBHS Tagger 0.0, GBHS Tagger 0.3, GBHS Tagger 0.5, GBHS Tagger 0.7.

Al ejecutar los algoritmos sobre el pequeño conjunto de datos se pudo apreciar que era necesario utilizar división de tareas, de tal forma, que la ejecución de los algoritmos tomara menos tiempo. En la siguiente sección, se describe la solución desarrollada.

Con la primera versión del etiquetador GBHS Tagger se ejecutaron los experimentos sobre todo el corpus Brown, obteniendo muy buen desempeño por parte del etiquetador propuesto, específicamente GBHS Tagger 0.5, es decir, con un valor para el parámetro de optimización de 0.5. Cabe aclarar que este algoritmo utilizó una ventana de tres, es decir, se consideró como contexto la etiqueta de la palabra predecesora y la palabra sucesora.

En tercera instancia, se desarrolló una segunda versión que involucró la inicialización mejorada (presentada en el Algoritmo 4.3), que incluye el uso del parámetro Alpha y el optimizador local con los mismos valores para el parámetro de optimización de la primera versión. Para definir el valor del parámetro Alpha también se realizaron evaluaciones sobre un pequeño conjunto de datos, obteniendo que el valor que más

se ajustaba fue 0.5. Esta versión del algoritmo memético de etiquetado se llama GBHS Tagger2, para efectos de la experimentación, se llamó así: GBHS Tagger2 0.0, GBHS Tagger2 0.3, GBHS Tagger2 0.5, GBHS Tagger2 0.7. Se ejecutaron experimentos con un conjunto pequeño de datos que permitieron apreciar que GBHS Tagger2, obtenía un mejor desempeño en comparación con la primera versión.

En cuarta instancia, se realizó una pequeña modificación a GBHS 2 Tagger, que buscaba combinar la inicialización aleatoria y la inicialización mejorada de la memoria armónica, incluyendo también el uso del optimizador local con los mismos valores para el parámetro de optimización. Esta versión se llamó GBHS Tagger3, para efectos de la expermientación, se llamó así: GBHS Tagger3 0.0, GBHS Tagger3 0.3, GBHS Tagger3 0.5, GBHS Tagger3 0.7.

Teniendo estas tres versiones del algoritmo memético de etiquetado se procedió a:

1. Realizar pruebas con ventanas de contexto diferentes a trigramas, como pentagramas, es decir, teniendo en cuenta las etiquetas de las dos palabras predecesoras y las dos palabras sucesoras. Los experimentos se aplicaron sobre un conjunto pequeño de datos del corpus, obteniendo mejores resultados para el caso de los trigramas, por tanto, se definió como ventana de contexto para el algoritmo memético propuesto el trigrama, la cual como se describió en la sección 4.2, es la que se utiliza para calcular la función objetivo del algoritmo.

2. Construir las otras versiones del etiquetador HSTagger 2 y HSTagger, descritas en el siguiente capítulo, las cuales constituyeron la línea base para la experimentación.

3. Se procedió a ejecutar los experimentos sobre el corpus completo en donde se incluyeron:

    - Las tres versiones del algoritmo memético de etiquetado propuesto (GBHS Tagger, GBHS Tagger2, GBHS Tagger3) incluyendo el uso de los diferentes valores de optimización.

    - Las tres versiones del algoritmo HSTagger (HSTagger, HSTagger2, HSTagger3).

- El algoritmo aleatorio de etiquetado (Azar).

La descripción de los experimentos y los resultados obtenidos se presentan en el siguiente capítulo.

4.4.2 Sobre la implementación del algoritmo

Se considera relevante mencionar algunos aspectos que se tuvieron en cuenta al momento de la implementación del algoritmo GBHS Tagger propuesto (en sus tres versiones y de los algoritmos de la línea base presentados en la siguiente sección) como son:

- Para ejecutar los experimentos, se implementó la ejecución distribuida de tareas de etiquetado, de tal forma, que fue posible configurar la ejecución de los experimentos utilizando 20 computadores, en donde cada computador se conecta a un servidor de base de datos (ubicado en la nube), el cual asigna la siguiente tarea para su ejecución. Cada tarea corresponde a las 30 ejecuciones del algoritmo sobre una oración, y estas 30 tareas se ejecutan en un mismo computador usando múltiples hilos de ejecución de acuerdo con el número de núcleos disponibles en el procesador. Al finalizar la tarea, el cliente (cada computador) registra los resultados promedios de las 30 ejecuciones del algoritmo sobre cada oración (cada ejecución sobre una oración corresponde a 110 evaluaciones de la función objetivo, EFOS). Esto permitió que la ejecución de los algoritmos se realizara en un tiempo considerablemente menor (días), al que hubiese tomado realizar la ejecución de manera secuencial sobre una sola computadora (meses).

- Un servidor de base de datos que se encuentra en la nube se encarga únicamente de distribuir tareas y almacenar resultados que luego se usan para el calculo de las estadísticas del rendimiento de cada uno de los algoritmos. Por otro lado, los equipos cliente cuentan con el algoritmo y una base de datos propia en la cual se almacena el corpus etiquetado con sus correspondientes estadísticas. Con el objetivo de hacer más eficiente la ejecucipon del algoritmo, la evaluación de la función objetivo (conforme se muestra en la armonía o solución de la **Figura 4.1**) se realiza a través de un único acceso a la base de datos. En dicho llamado se recupera la probabilidad de etiqueta y la probabilidad de trigramas por separado para cada palabra de la oración, evitando consultarla cada vez que se requiera algún valor. Esto permite, que el optimizador local use información que ya se

encuentra en la armonía (solución) pueda decidir la palabra a la cual se le debe modificar la etiqueta.

- La información de probabilidades de las etiquetas posibles de cada palabra y trigramas posibles en los datos de entrenamiento se obtuvo mediante consultas estándar SQL y se almacenó en tablas de la base de datos del corpus.

- La memoria tabú que se utiliza en el algoritmo se implementó mediante una tabla Hash que almacena explicamente la solución generada en el orden de las palabras que hacen parte de la oración.

4.4.3 Sobre la complejidad del algoritmo

El algoritmo memético de etiquetado propuesto, realiza varias iteraciones las cuales están determinadas de manera general, por el tamaño de la memoría armónica (HMS), el número de improvisaciones (NI), la probabilidad de ejecutar el optimizador local ($ProbOpt$) y el número de optimizaciones locales a realizar ($MaxNeighbors$). Sin importar estos valores, el algoritmo mide su complejidad por el número de evaluaciones de la función objetivo (EFOs) que ejecuta, que para el caso de los experimentos fue fijada en 110. La evaluación de cada EFO implica el calculo de la probabilidad de las N etiquetas de las palabras en la oración y sus correspondientes trigramas. Estas dos tareas se realizan en $O(4 * accesosBD)$ que corresponde al acceso al índice de la base de datos y al acceso al dato propiamente almacenado en la tabla correspondiente, tarea que se realiza para la probabilidad de la etiqueta como para el trigrama. Teniendo en cuenta que son N palabras, el calculo de la EFO es de $O(4 * accesosBD * N) \approx O(N * accesosBD)$. Es así como la complejidad del algoritmo se puede resumir en la siguiente expresión $O(EFOs * N * accesosBD)$. Si una oración tiene 20 palabras y el algoritmo se ejecuta con 110 EFOs, el costo de la ejecución del algoritmo propuesto esta acotado por $O(2000\, accesosBD)$ y sería exactamente de 8000 accesos a la base de datos.

4.5 Síntesis

En la revisión realizada en el capitulo 2, se pudo apreciar que los algoritmos metaheurísticos, ofrecen poderosas perspectivas para la construcción de etiquetadores. La propuesta descrita en el presente capitulo, involucra el uso de un algoritmo memético, el cual tiene como base la metaheurística GBHS junto con la

inclusión de conocimiento del problema a través de un optimizador local basado en Hill Climbing y el uso de una memoría tabu explícita, lo cual favorece el desempeño del algoritmo propuesto, conviritiéndose en el primer algoritmo memético diseñado y utilizado explícitamente para el problema de etiquetado.

Una de las contribuciones más significativas que surge de este estudio es que al incluir conocimiento del problema a la identificación de partes del discurso, a través de la utilización del optimizador local basado en la metaheurística subiendo la colina (Hill Climbing), se pueden obtener mejores resultados como se muestra en el siguiente capitulo.

Capítulo 5

Marco Experimental

En este capitulo se presentan los experimentos desarrollados con el fin de evaluar el desempeño del algoritmo propuesto en sus diferentes versiones, en contraste con los algoritmos desarrollados como línea base. En la sección 5.1, se presenta el experimento desarrollado con el corpus Brown seleccionado, para el caso de estudio de inglés. En la sección 5.2, se realizan dos experimentos para el caso de la lengua nasa yuwe utilizando el corpus etiquetado construido para esta lengua. Al final de cada una de las secciones, se presenta una síntesis de los resultados obtenidos.

El proceso seguido para desarrollar los experimentos se resume en la **Figura 5.1**, en donde a partir del corpus etiquetado (inglés con el corpus Brown o con el corpus Nasa yuwe), se obtienen las oraciones etiquetadas usando el conjunto de etiquetado universal (tagset), de allí, se obtienen los conjuntos de datos para entrenamiento y prueba (con el método de validación cruzada), se aplica el etiquetador a los datos de prueba y se obtienen las oraciones etiquetadas, a las cuales se les aplica la medida de precisión (ver **Ecuación 5. 1**) para conocer el desempeño del etiquetador utilizado en el experimento.

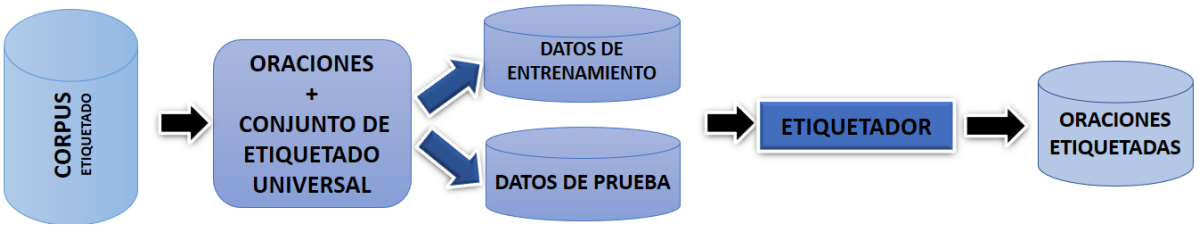

Figura 5.1. Representación del proceso utilizado para el desarrollo de los experimentos. **Fuente**: Propia

5.1 Caso de estudio 1: Corpus Brown (inglés)

5.1.1 Preprocesamiento del corpus

Como se señaló en la **sección 3.2**, el corpus seleccionado para el caso de estudio de inglés fue el Corpus Brown [108].

El corpus original se encuentra almacenado en texto plano, en donde cada palabra dentro de una oración aparece con su correspondiente etiqueta, como se muestra en la **Figura 3.1**, de la forma: **palabra / etiqueta Brown**, por tanto, para facilitar el uso de este corpus por los diferentes etiquetadores con los que se realizó el experimento, se realizaron algunos pasos previos, a saber:

- Se identificó cada una de las frases y se realizó su correspondiente sistematización en una base de datos, en la cual cada frase se identificó, se generó la frase en texto, es decir, sola sin la etiqueta y la frase etiquetada, logrando consolidar 52,998 frases u oraciones.

- Como se ha mencionado antes, el conjunto de etiquetado universal propuesto por Petrov [10], facilita el proceso de etiquetado y también es muy utilizado en la literatura, por tanto, se realizó la equivalencia entre las 472 etiquetas Brown con las 12 etiquetas del conjunto de etiquetado universal. Esta tabla de equivalencia se obtuvo de NLTK Project [15]. La **Tabla 5.1**, muestra algunas de estas equivalencias. En la **sección 2 del Anexo B**, se encuentra el detalle de las equivalencias.

Tag BROWN	Tag UNIVERSAL	Tag BROWN	Tag UNIVERSAL
(	.	Dt	det
(-hl	.	dt$	det
)	.	In	adp
)-hl	.	in+in	adp
*	Adv	Jj	adj
*-hl	Adv	jj$-tl	prt
*-nc	Adv	jj+jj-nc	adj
*-tl	Adv	nn+nn-nc	noun
,	.	nn-hl	noun

Tabla 5.1. Ejemplo de mapeo etiquetas Brown – etiquetado Universal. **Fuente:** *[15]*

- Se sistematizó y aplicó el mapeo de etiquetado Universal a las oraciones identificadas anteriormente, de tal forma, que se obtuvo una tabla con la estructura mostrada en **Tabla 5.2**, la cual facilitó el procesamiento y la extracción de

información del corpus durante el proceso de etiquetado, a través de procedimientos almacenados para cada trigrama presente en el corpus.

- Para el desarrollo de los experimentos se utilizó validación cruzada de 5 carpetas (5-folders), por tanto, se asignó la respectiva carpeta (folder) a cada oración. En la **Tabla 5.2** se muestra la oración 1, con su identificador (IDFrase), el identificador de la palabra (IDTermino), la palabra (Termino), el orden de la palabra en la oración (Orden), la etiqueta asignada a la palabra (Etiqueta), el identificador de la etiqueta de la palabra predecesora (ID Etiqueta Predecesora o -1 cuando no se tiene palabra predecesora), el identificador de la etiqueta asignada a la palabra (ID Etiqueta), el identificador de la etiqueta sucesora (ID Etiqueta Sucesora o -1 cuando no se tiene palabra sucesora) y el folder asignado a la oración (Folder), que es un valor de 1 a 5. Los campos ID Etiqueta Predecesora - ID Etiqueta - ID Etiqueta Sucesora, conforman el contexto de la palabra a etiquetar, es decir, el trigrama, del que se habla en el capitulo 4.

ID Frase	ID Termino	Termino	Orden	Etiqueta	ID Etiqueta Predecesora	ID Etiqueta	ID Etiqueta Sucesora	Folder
1	1	the	1	det	-1	7	2	1
1	2	fulton	2	noun	7	2	2	1
1	3	county	3	noun	2	2	3	1
1	4	grand	4	adj	2	3	2	1
1	5	jury	5	noun	3	2	1	1
1	6	said	6	verb	2	1	2	1
1	7	friday	7	noun	1	2	7	1
1	8	an	8	det	2	7	2	1
1	9	investigation	9	noun	7	2	5	1
1	10	of	10	adp	2	5	2	1
1	11	atlanta’s	11	noun	5	2	3	1
1	12	recent	12	adj	2	3	2	1
1	13	primary	13	noun	3	2	2	1
1	14	election	14	noun	2	2	1	1
1	15	produced	15	verb	2	1	11	1
1	16	“	16	.	1	11	7	1
1	17	no	17	det	11	7	2	1
1	18	evidence	18	noun	7	2	11	1
1	19	“	19	.	2	11	5	1
1	20	that	20	adp	11	5	7	1
1	21	any	21	det	5	7	2	1
1	22	irregularities	22	noun	7	2	1	1
1	23	took	23	verb	2	1	2	1
1	24	place	24	noun	1	2	11	1
1	25	.	25	.	2	11	-1	1

Tabla 5.2. Ejemplo de tabla de relaciones de la sistematización del Corpus Brown. **Fuente:** Propia

5.1.2 Algoritmos implementados para la hacer la comparación de resultados

Se realizaron varias implementaciones para constratar los resultados obtenidos por el etiquetador propuesto (GBHS Tagger) así:

- Tres versiones de HSTagger [19, 98, 99], las cuales demostraron buenos resultados en contraste con otros métodos reconocidos de etiquetado como Modelos ocultos de Markov, Modelo de máxima entropía y una versión del modelo de Brill. Las versiones son:

    - HSTagger, es un algoritmo basado en búsqueda armónica (Harmony Search, HS), el cual tiene inicialización aleatoria para la memoria armónica.

    - HSTagger 2, es un algoritmo también basado en búsqueda armónica al cual se le ha introducido una inicialización mejorada donde se crea la memoria armónica a partir de las etiquetas de las palabras con mayor probabilidad de manera similar a lo explicado en el **Algoritmo 4.2**, también tomando en cuenta el parámetro Alpha.

    - HSTagger 3, también es basado en HS e involucra el uso de una inicialización mejorada de la memoria armonica (**Algoritmo 4.2**), adicionando una modificación al momento de crear el improviso con el parámetro HCMR, que utiliza las palabras con el mayor número de ocurrencias de cada palabra en la memoria armónica, que han sido calculadas previamente.

- Un algoritmo aleatorio de etiquetado, es decir, el cual genera nuevas soluciones aleatorias para el etiquetado de las palabras en una oración.

- Adicionalmente, se utilizó NLTK software [15] para evaluar los algoritmos HMM, TnT y Perceptrón multicapa utilizando igualmente el corpus Brown como datos de entrenamiento y pruebas, obteniendo valores muy similares entre ellos tres, por lo tanto, en el siguiente experimento solo se presentan los resultados con TnT.

5.1.3 Configuración del experimento

1. **Conjunto de Datos:** La **Tabla 5.3**, muestra la distribución de las oraciones y palabra en cada uno de los conjuntos de datos (data sets) de entrenamiento y

prueba, por ejemplo, si los datos de prueba son las oraciones de la carpeta (folder) 1, entonces los datos de entrenamiento son las oraciones contenidas en las carpetas 2 a 5 y así sucesivamente para las otras carpetas.

Datos de prueba Folder	Oraciones en los datos de prueba	Palabras en los datos de prueba	Folders con datos de entrenamiento	Palabras en los datos de entrenamiento	Palabras comunes	Palabras desconocidas
1	10595	23105	2,3,4,5	45113	18398	4707 (20.4%)
2	10600	22852	1,3,4,5	45199	18231	4621 (20.2%)
3	10600	23130	1,2,4,5	45009	18319	4811 (20.8%)
4	10600	22929	1,2,3,5	45130	18239	4690 (20.5%)
5	10603	23111	1,2,3,4	45025	18316	4795 (20.8%)

Tabla 5.3. Data sets de prueba y entrenamiento usados para los experimentos. **Fuente:** Propia

2. **Medida de evaluación:** Como medida de evaluación de desempeño de los algoritmos, al momento de la comparación, se estableció el valor de la precisión según se presenta en la **Ecuación 5. 1** la cual es utilizada en los diferentes trabajos del estado del arte. Para todos los algoritmos se calcula el promedio de la precisión obtenida en todos los conjuntos de datos de prueba sobre un número de ejecuciones.

$$Precision\ = \frac{(\#\ palabras\ correctamente\ etiquetadas)}{(\#\ palabras)} * 100$$

Ecuación 5. 1. Formula Precisión. Fuente: [19]

3. **Número de ejecuciones:** Cada algoritmo se ejecutó 30 veces (Teorema del Limite Central) sobre cada oración y sobre estos fueron calculados los valores de precisión promedio y desviación estándar para cada oración. Todos los algoritmos se ejecutaron con un máximo de 110 evaluaciones de la función objetivo para cada oración.

4. **Parámetros de los algoritmos HSTagger:** Los parámetros utilizados para las versiones de HSTagger fueron: HMS= 20, HMCR= 0.65 y PAR= 0.25. Estos valores fueron los recomendados por los autores del etiquetador.

5. **Parámetros de los algoritmos GBHS Tagger:** Los parámetros utilizados para las versiones de GBHS Tagger fueron: HMS= 10, HMCR= 0.95, PARMin= 0.01,

PARMax= 0.99, MaxNeighbors= 5, Alfa=0.5 y ProbOpt = 0.0, 0.3, 0.5, 0.7, valores que permitieron comparar GBHS Tagger sin utilizar optimización local y con diferentes valores de probalidad.

6. **Algunos aspectos relevantes sobre la implementación de los algoritmos:** Para mejorar los desempeños a nivel de ejecución de los algoritmos se incluyeron los siguientes detalles:

- Sobre la **Tabla 5.2**, mencionada en la **sección 5.1.1**, en donde reposan las relaciones del corpus etiquetado, se elaboraron varios procedimientos almacenados para recuperar la información de la base de datos y realizar algunos cálculos disminuyendo los tiempos de acceso por parte de los algoritmos a la base de datos. Los procedimientos utilizados permiten obtener: las posibles etiquetas de un término, la etiqueta más probable de un término, la probabilidad de la etiqueta de un término con respecto al conjunto de datos de entrenamiento y las oraciones a etiquetar.

5.1.4 Resultados

En la **Tabla 5.4**, se muestra el desempeño de los algoritmos que se han ejecutado bajo las condiciones descritas anteriormente. El algoritmo memético de etiquetado propuesto, GBHS Tagger con optimización local (en sus versiones 2 y 3), presenta los mejores resultados de precisión que los otros algoritmos: etiquetador aleatorio (Azar), TnT, HSTagger (en todas sus versiones) y GBHS Tagger sin optimización local.

En la **Tabla 5.4**, también se puede apreciar el desempeño de los etiquetadores en las oraciones que contienen palabras desconocidas (unknown words), en donde GBHS Tagger en las versiones con optimización local presenta mejores valores de precisión, sin embargo, GBHS Tagger 2 obtiene los mejores resultados.

Algoritmos	Parámetro (ProbOpt)	Número de oraciones	Precisión (%)	Desviación Estándar	Precision(%) Palabras desconocidas	Desviación estándar
Azar	-	52998	82.4409	0.8313	79.9964	0.0074
TnT	-	52998	84.2699	4.2428	80.9674	0.0225
HSTAGger	-	52998	91.5903	0.3232	88.8583	0.0028
HSTAGger2	-	52998	93.5952	0.0931	90.2307	0.0013
HSTAGger3	-	52998	92.2751	0.0511	86,6693	0,0017
GBHS Tagger	0.0	52998	92.2474	0.3820	89.7843	0.0035
GBHS Tagger	0.3	52998	93.4417	0.2516	90.8924	0.0024
GBHS Tagger	0.5	52998	93.4444	0.2512	90.8959	0.0024
GBHS Tagger	0.7	52998	93.4414	0.2514	90.8923	0.0024
GBHS Tagger 2	0.0	52998	94.5615	0.0481	91.6568	0.0008
GBHS Tagger 2	0.3	52998	**95.1959**	**0.0314**	**92.8542**	**0.0005**
GBHS Tagger 2	0.5	52998	**95.1959**	**0.0314**	**92.8542**	**0.0005**
GBHS Tagger 2	0.7	52998	**95.1959**	**0.0314**	**92.8542**	**0.0005**
GBHS Tagger 3	0.0	52998	94.4279	0.0654	91,4607	0,0011
GBHS Tagger 3	0.3	52998	95.1213	0.0329	92,6735	0,0006
GBHS Tagger 3	0.5	52998	95.1213	0.0329	92,6735	0,0006
GBHS Tagger 3	0.7	52998	95.1213	0.0329	92,6735	0,0006

Tabla 5.4. Resultados de la ejecución de algoritmos. Los mejores resultados se muestran en negrilla.

En la **Figura 5.2**, se puede apreciar el desempeño de cada uno de los algoritmos evaluados, observando una diferencia significativa en los valores de precisión alcanzado por los diferentes algoritmos, confirmando que el algoritmo GBHS Tagger 2, se desempeña mejor, seguido por el algoritmo GBHS Tagger 3.

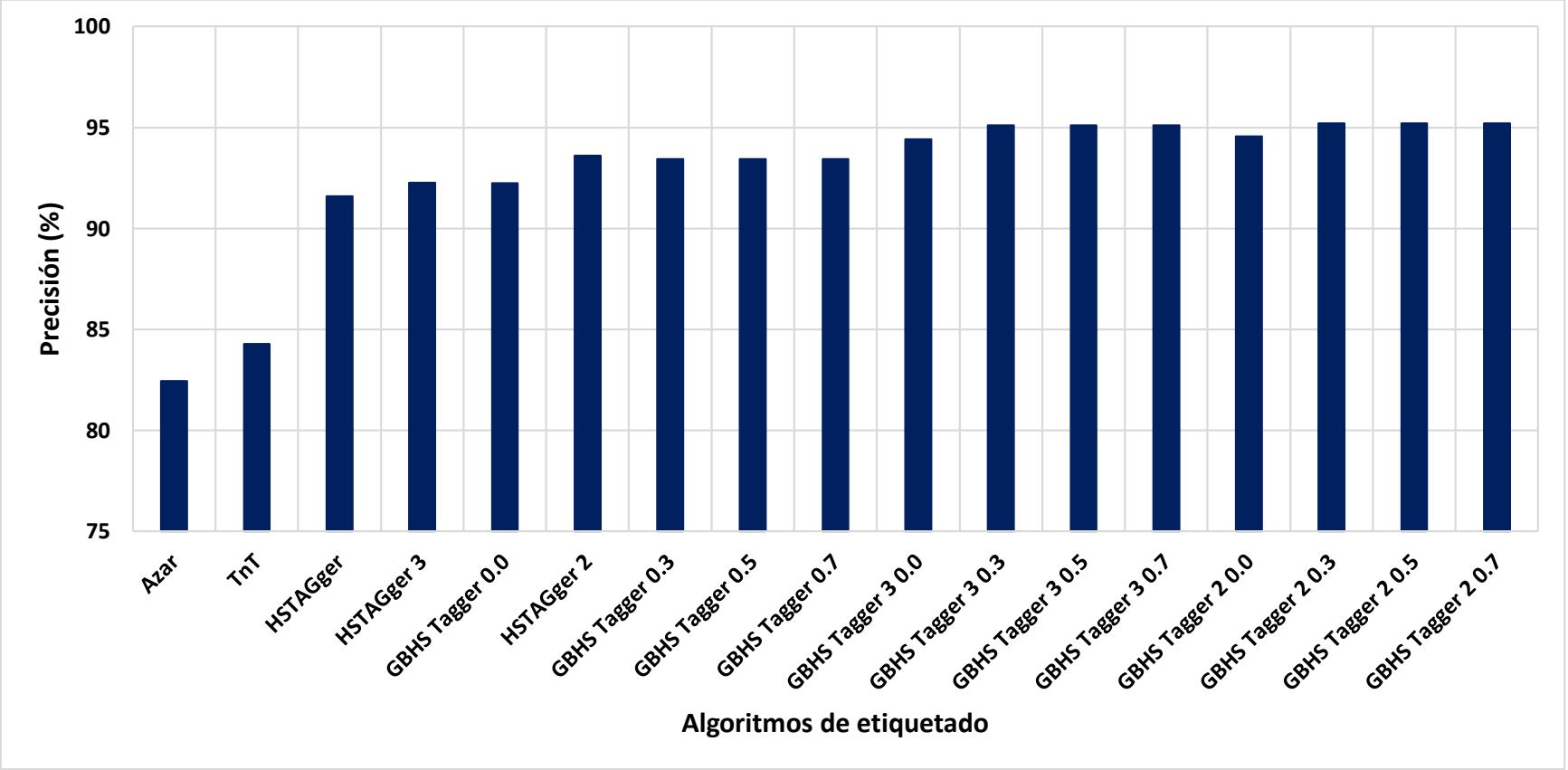

Figura 5.2. Comparación de desempeño de los algoritmos.

La **Tabla 5.5**, muestra los valores de precisión para cada algoritmo en cada folder de validación. En todas las oraciones evaluadas, el algoritmo de etiquetado que reporta mejores valores de precisión es el GBHS Tagger 2 con optimización local e inicialización mejorada. El algoritmo de etiquetado aleatorio (Azar) es el que presenta menor desempeño de todos los folders y el algortimo TnT es el que presenta mayor desviación estándar.

Los resultados de GBHS Tagger 2 con probabilidad de optimización de 0.3 en adelante son mejores que el algoritmo sin utilizar el optimizador local, lo mismo sucede con las versiones del algoritmo GBHS Tagger y GBHS Tagger 3. Resultados que confirman que el uso del optimizador local con probabilidad de 0.3 en adelante, junto con el uso de la inicialización mejorada de la memoria armónica mejora los resultados en el algoritmo propuesto. También se puede apreciar que el algoritmo etiquetador HSTagger 2 es mejor que HSTagger 3, HSTagger, TnT y Azar.

El desempeño de TnT se ve afectado por el proceso de análisis léxico (tokenizers) manual del corpus Brown, mientras que los algoritmos metaheurísticos no. Por ejemplo, en la oración “... the recent atlanta’s investigation ...”, TnT etiqueta el token “atlanta’s” incorrectamente, debido a que en el corpus este no fue dividido en dos tokens: “atlanta” and “’s”. La misma situación afecta el desempeño de los otros etiquetadores en NLTK tales como HMM y Perceptrón.

Con el fin de hacer un análisis a los resultados obtenidos, el test no paramétrico de Friedman y de Wilcoxon fueron aplicados a estos datos, los cuales permitieron establecer que las diferencias entre los algoritmos son estadísticamente significativas:

Algoritmos	Folders					Total	Desv Estand	Promedio de tiempo (Segundos)
	1	2	3	4	5			
GBHS Tagger 3 con 0.7	95.1421	95.0890	95.1461	95.1550	95.0747	95,1213	0,0329	24.9762
GBHS Tagger 3 con 0.5	95.1421	95.0890	95.1461	95.1550	95.0747	95,1213	0,0329	25.0132
GBHS Tagger 3 con 0.3	95.1421	95.0890	95.1461	95.1550	95.0747	95,1213	0,0329	25.5065
GBHS Tagger 3 con 0.0	94.5031	94.3416	94.3829	94.4706	94.4413	94,4279	0,0585	24.8591
GBHS Tagger 2 con 0.7	**95.2239**	**95.1787**	**95.2025**	**95.2299**	**95.1444**	**95.1959**	**0.0314**	25.2849
GBHS Tagger 2 con 0.5	**95.2239**	**95.1787**	**95.2025**	**95.2299**	**95.1444**	**95.1959**	**0.0314**	24.8055
GBHS Tagger 2 con 0.3	**95.2239**	**95.1787**	**95.2025**	**95.2299**	**95.1444**	**95.1959**	**0.0314**	24.8008
GBHS Tagger 2 con 0.0	94.6291	94.4980	94.5209	94.5980	94.5615	94.5615	0.0481	23.7773
GBHS Tagger con 0.7	93.6769	93.0373	93.2665	93.5519	93.6747	93.4414	0.2514	**10.6113**
GBHS Tagger con 0.5	93.6805	93.0410	93.2689	93.5547	93.6768	93.4444	0.2512	12.0091
GBHS Tagger con 0.3	93.6769	93.0373	93.2665	93.5519	93.6758	93.4417	0.2516	11.4179
GBHS Tagger con 0.0	92.6221	91.6444	91.9596	92.4364	92.5747	92.2474	0.3820	**10.9308**
HSTAGger 3	92.2904	92.3189	92.2742	92.3140	92.1782	92.2751	0.0511	22.3344
HSTAGger 2	93.7491	93.5258	93.4864	93.6443	93.5952	93.5952	0.0931	22.9612
HSTAGger	91.9486	91.0889	91.3380	91.8070	91.7693	91.5903	0.3232	12.0829
TnT	83.5415	91.9662	78.8870	83.3974	83.5573	84.2698	4.2428	13,7691
Azar	83.3960	81.2045	81.7253	82.8884	82.9901	82.4409	0.8313	11,7691

Tabla 5.5. Precision obtenido por folder. **Fuente:** Propia

1. **Test de Friedman:** La **Tabla 5.6**, muestra los puntajes obtenidos por cada algoritmo, una vez se ha aplicado el Test de Friedman de NXN en donde se puede apreciar que el mejor algoritmo de etiquetado es el GBHS Tagger 2 con optimización local igual a 0.3, 0.5 y 0.7. Este test tiene 16 grados de libertad (78.3216) y un valor de p (p-value) de $4.0426E^{-10}$. Este valor de p hace el ranking estadísticamente significativo, ya que es menor que 0.05.

Algoritmo	Ranking Friedman
GBHS Tagger 2 con 0.3	2
GBHS Tagger 2 con 0.5	2
GBHS Tagger 2 con 0.7	2
GBHS Tagger 3 con 0.3	5
GBHS Tagger 3 con 0.5	5
GBHS Tagger 3 con 0.7	5
GBHS Tagger 2 con 0.0	7
GBHS Tagger 3 con 0.0	8
HSTagger 2	9.6
GBHS Tagger con 0.5	9.8
GBHS Tagger con 0.3	11.2
GBHS Tagger con 0.7	11.4
GBHS Tagger con 0.0	13.6
HSTagger 3	13.6
HSTagger	15.2
TnT	15.8
Azar	16.8

Tabla 5.6. Resultados Test de Friedman. **Fuente:** Propia con software KEEL *[126]*

2. **Test de Wilcoxon:** La aplicación de este mostró con un 90% de confianza que los resultados obtenidos con los algoritmos GBHS Tagger 2 con valores de optimización de 0.3, 0.5 y 0.7 son mejores que los de los otros algoritmos con los que se comparó. Adicionalmente se pudo observar que:

   - HSTagger 2 supera a HSTagger 3 y GBHS sin optimización local (con probabilidad de optimización 0.0).

   - Los resultados del etiquetador aleatorio (Azar) son superados por todos los otros algoritmos de etiquetado, seguido por TnT y HSTagger.

5.1.5 Medidas obtenidas de la matriz de confusión

Teniendo en cuenta el conjunto de etiquetas (clases) utilizado, es posible realizar una comparación entre ellas, la cual se ha realizado con las medidas micro ponderadas de precisión, recuerdo y medida F, obtenidas de una matriz de confusión desarrollada

para el mejor algoritmo de etiquetado (GBHS Tagger 2 para valores de optimización de 0.3). Los resultados obtenidos se presentan en la **Tabla 5.7**.

Etiqueta	Medidas Micro		
	Precisión (%)	Recuerdo (%)	Medida F (%)
verb	95,9415	94,2506	95,0885
noun	94,3079	95,2510	94,7771
adj	91,2595	87,8909	89,5435
adv	90,2810	87,7181	88,9811
adp	92,6145	96,5114	94,5228
conj	99,2057	98,6483	98,9262
det	95,9480	98,7999	97,3531
num	99,3514	80,9518	89,2128
prt	87,4846	81,4357	84,3519
X	83,7264	51,7460	63,9615
.	99,1835	99,9899	99,5851
pron	96,3864	95,1215	95,7498
Promedio	94,8513	94,8595	94,8194

Tabla 5.7. Medidas Matriz de confusión experimento sobre corpus Brown (inglés*)*

Se puede apreciar que los valores máximos de precisión son para las etiquetas conj (conjunciones), num, puntuación y pron (pronombres), y los menores valores son para X (otras palabras) y prt (partículas).

5.1.6 Síntesis

Los experimentos permitieron apreciar que el algoritmo GBHS Tagger propuesto en sus diferentes versiones mostró ser una solución adecuada para el problema de etiquetado, al obtener resultados sobresalientes en su desempeño en comparación con los algoritmos evaluados.

El algorimo GBHS Tagger incluye tres parámetros adicionales:

- Uno que controla la probilidad de uso del optimizador local, alcanzando los mejores resultados en su primera versión cuando la probabilidad es 0.5 sin la mejora en la inicialización de la memoria armónica y en la segunda versión cuando la probabilidad es de 0.5 en adelante, usando la mejora en la inicialización de la memoria armónica.

- Un segundo parámetro, que controla la cantidad de vecinos que se explotan sobre la mejor armonía en la memoria armónica al aplicar el optimizador local.

- Un tercer parámetro, que controla si se utiliza o no la mejora en la inicialización de la memoria armónica.

El afinamiento de estos parámetros se realizó durante el diseño del algoritmo, el cual se basó en la experiencia y en los resultados obtenidos en el desempeño del algoritmo, no obstante, cabe resaltar, que es preciso como trabajo futuro realizar un proceso formal de afinamiento de estos parámetros lo cual se podría realizar mediante el uso de: arreglos de cobertura (Covering arrays), una metaheurística o la ejecución del algoritmo con diferentes combinaciones en los valores de los parámetros (malla), siendo este último un proceso dispendioso y costoso en tiempo de ejecución.

Otro resultado importante de esta sección es la publicación de un artículo titulado “Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging” [127] en The Fifth International Conference on Mining Intelligence and Knowledge Exploration (MIKE 2017), realizada en diciembre 13 - 15, 2017, en IDRBT, Hyderabad, India (DOI https://doi.org/10.1007/978-3-319-71928-3_20). Los Proceedings fueron publicados en un ejemplar de la revista Lecture Notes in Artificial Intelligence en Springer con ISSN 0302-9743 ISSN 1611-3349 (electronic), ISBN 978-3-319-71927-6 ISBN 978-3-319-71928-3 (eBook), https://doi.org/10.1007/978-3-319-71928-3. Esta publicación se encuentra indexada por el PUBLINDEX de Colciencias en la categoría A2. En el Anexo C se encuentra el artículo publicado.

5.2 Caso de estudio 2: Corpus nasa yuwe

Se realizaron dos experimentos con el corpus etiquetado para nasa yuwe, descrito en el capitulo 3, el primero, utilizando validación cruzada con k=10 folders y el segundo, utilizando validación cruzada dejando un registro fuera para validación (leave-one-out). El procedimiento utilizado es el mismo presentado al inicio de este capítulo.

5.2.1 Configuración de los experimentos

1. **Algoritmos de etiquetado utilizados:** Se han utilizado los mismos algoritmos descritos en el experimento desarrollado con el corpus Brown del idioma inglés, como son: las tres versiones de HSTagger [19, 98, 99], el algoritmo aleatorio de etiquetado (previamente descritos en la sección 5.1.3) y las tres versiones del algoritmo memético GBHS Tagger propuesto en el capitulo 4.

2. **Parámetros de los algoritmos HSTagger:** Los parámetros utilizados fueron: HMS= 20, HMCR= 0.65 y PAR= 0.25.

3. **Parámetros de los algoritmos GBHS Tagger:** Los parámetros utilizados fueron: HMS= 10, HMCR= 0.95, PARMin= 0.01, PARMax= 0.99, MaxNeighbors= 5, Alfa=0.5 y ProbOpt = 0.0, 0.3, 0.5, 0.7.

4. **Número de ejecuciones:** En los dos experimentos, cada algoritmo fue ejecutado 30 veces sobre cada oración y sobre estos resultados fueron calculados los valores de precisión para cada algoritmo. Al igual que con el experimento del corpus Brown, cada algoritmo se ejecutó un máximo de 110 evaluaciones de la función objetivo por cada oración. La precisión de los algoritmos se calculó como el promedio de la precisión de cada folder, utilizando la **Ecuación 5. 1**.

5.2.2 Preprocesamiento del corpus

Al momento de hacer la sistematización del corpus lingüístico etiquetado para nasa yuwe, mostrado en el capítulo 3, se obtuvo una tabla de relaciones similar a la construída para el corpus Brown (ver **Tabla 5.2**). Para efectos del corpus etiquetado para nasa yuwe, en la **Tabla 5.8**, se presentan estas relaciones, donde se encuentra el identificador de la oración (Id_Frase), el identificador de la palabra (Id_Termino), el número de la posición de la palabra en la oración (Orden), el identificador de la etiqueta Petrov con la que se ha etiquetado la palabra (Id_Etiqueta), los identificadores de las etiquetas Petrov con las que se han etiquetado las palabras predecesora (Id_Etiq_Pred o -1 si no hay etiqueta predesecora) y sucesora (Id_Etiq_Suc o -1 si no hay etiqueta sucesora) a la que se desea etiquetar, es decir, el mismo trigrama, del que se habla en el capitulo 4 y la sección anterior, al momento de representar la solución para el algoritmo propuesto GBHS Tagger.

Id_Frase	Id_Termino	Orden	Id_Etiq_Pred	Id_Etiqueta	Id_Etiq_Suc	Folder
1	1	1	-1	2	2	1
1	2	2	2	2	2	1
1	3	3	2	2	-1	1
2	4	1	-1	4	2	2
2	5	2	4	2	3	2
2	6	3	2	3	1	2
2	7	4	3	1	4	2
2	8	5	1	4	2	2
2	9	6	4	2	1	2
2	10	7	2	1	3	2
2	11	8	1	3	3	2
2	12	9	3	3	2	2
2	13	10	3	2	2	2
2	14	11	2	2	1	2
2	15	12	2	1	11	2
2	16	13	1	11	-1	2

Tabla 5.8. Ejemplo de tabla de relaciones de la sistematización del Corpus Nasa Yuwe. **Fuente:** Propia

5.2.3 Experimento 1 (10 folders)

1. **Conjunto de datos:** La **Tabla 5.9**, muestra la cantidad de oraciones en cada conjunto de datos de entrenamiento y prueba al dividir el corpus en 10 folders, implicando que si las oraciones del folder 1 son tomadas como datos de prueba (test data), el conjunto de datos de entrenamiento (training data) esta compuesto por las oraciones que se encuentran en las carpetas (folders) 2 a 10, y así sucesivamente, para las otras carpetas.

Conjunto de datos de prueba	Oraciones en los datos de prueba	Palabras en los datos de prueba	Folders con datos de entrenamiento	Palabras en los datos de entrenamiento	Palabras comunes	Palabras desconocidas
1	18	197	2,3,4,5,6,7,8,9,10	1805	109	88 (44.67 %)
2	18	153	1,3,4,5,6,7,8,9,10	1849	86	67 (43.79 %)
3	18	179	1,2,4,5,6,7,8,9,10	1823	93	86 (48.04 %)
4	18	233	1,2,3,5,6,7,8,9,10	1769	113	120 (51.50 %)
5	18	229	1,2,3,4,6,7,8,9,10	1773	117	112 (48.91 %)
6	17	198	1,2,3,4,5,7,8,9,10	1804	102	96 (48.48 %)
7	17	249	1,2,3,4,5,6,8,9,10	1753	136	113 (45.38 %)
8	17	179	1,2,3,4,5,6,7,9,10	1823	98	81 (45.25 %)
9	17	194	1,2,3,4,5,6,7,8,10	1808	93	101 (52.06 %)
10	17	191	1,2,3,4, 5,6,7,8,9	1811	110	81 (42.41 %)

Tabla 5.9. Conjunto de datos de entrenamiento y prueba para k=10 folders. **Fuente:** Propia

Por tanto, como se puede apreciar en la **Tabla 5.9**, el porcentaje de palabras desconocidas es muy alto, lo cual se debe a que el corpus etiquetado para nasa yuwe es pequeño, esto afecta considerablemente el rendimiento de los algoritmos de etiquetadores, como se puede observar en la siguiente sección de resultados, a diferencia de lo observado en los experimentos realizados con el corpus Brown, que es mucho más grande.

2. Resultados

La **Tabla 5.10** muestra los valores de precisión y desviación estándar de los algoritmos, donde los mejores resultados se pueden apreciar en el algoritmo GBHS Tagger en su primera versión, es decir, sin optimizador local, y con inicialización aleatoria de la memoría armónica, seguido por GBHS Tagger con valores de optimización local, luego GBHS Tagger 3, con valores de optimización local.

Algoritmos	Parámetros (ProbOpt)	Número de oraciones	Precisión (%)	Desviación Estándar	Precision(%) Palabras desconocidas	Desviación estándar
Azar	-	175	53,862	3,427	23,893	1,728
HSTAGger	-	175	57,294	3,395	32,801	2,119
HSTAGger2	-	175	57,957	3,468	35,897	2,318
HSTAGger3	-	175	50,893	3,585	12,290	**0,875**
GBHS Tagger	0.0	175	63,536	2,842	39,873	1,878
GBHS Tagger	0.3	175	62,529	2,701	36,095	1,677
GBHS Tagger	0.5	175	62,529	2,701	33,398	1,551
GBHS Tagger	0.7	175	62,529	2,701	37,135	1,725
GBHS Tagger 2	0.0	175	**63,867**	2,884	**45,742**	1,352
GBHS Tagger 2	0.3	175	63,783	3,035	20.157	1,252
GBHS Tagger 2	0.5	175	63,783	3,035	19,070	1,185
GBHS Tagger 2	0.7	175	63,783	3,035	25,099	1,559
GBHS Tagger 3	0.0	175	63,614	**2,701**	39,907	1,756
GBHS Tagger 3	0.3	175	63,333	2,955	30,842	2,152
GBHS Tagger 3	0.5	175	63,333	2,955	41,936	2,925
GBHS Tagger 3	0.7	175	63,333	2,955	38,389	2,678

Tabla 5.10. Resultados de la ejecución de los algoritmos para k= 10 folders. **Fuente:** Propia

En la **Tabla 5.10**, también se puede apreciar el desempeño de los etiquetadores en las oraciones que contienen palabras desconocidas (unknown words), en donde GBHS Tagger 2 sin optimización local, presenta los mejores valores de precisión, sin embargo, GBHS Tagger 3 obtiene resultados muy cercanos a los mejores valores de precisión.

Algoritmos	Folders										Total	Desv Estand
	1	2	3	4	5	6	7	8	9	10		
GBHS Tagger 3 con 0.7	65,651	67,255	61,530	58,264	62,951	62,037	64,525	66,462	58,969	65,689	63,333	2,955
GBHS Tagger 3 con 0.5	65,651	67,255	61,530	58,264	62,951	62,037	64,525	66,462	58,969	65,689	63,333	2,955
GBHS Tagger 3 con 0.3	65,651	67,255	61,530	58,264	62,951	62,037	64,525	66,462	58,969	65,689	63,333	2,955
GBHS Tagger 3 con 0.0	66,091	66,645	61,858	59,611	62,018	61,684	65,154	66,797	60,034	66,248	63,614	2,701
GBHS Tagger 2 con 0.7	66,413	**68,279**	61,311	**59,861**	**62,982**	**63,047**	**65,033**	66,704	58,454	65,742	63,783	3,035
GBHS Tagger 2 con 0.5	66,413	**68,279**	61,311	**59,861**	**62,982**	**63,047**	**65,033**	66,704	58,454	65,742	63,783	3,035
GBHS Tagger 2 con 0.3	66,413	**68,279**	61,311	**59,861**	**62,982**	**63,047**	**65,033**	66,704	58,454	65,742	63,783	3,035
GBHS Tagger 2 con 0.0	**66,413**	67,516	**61,894**	59,069	62,615	62,576	64,980	**66,909**	**60,017**	**66,684**	**63,867**	2,884
GBHS Tagger con 0.7	63,875	66,449	61,876	57,694	61,514	61,162	62,784	65,978	59,107	64,852	62,529	2,701
GBHS Tagger con 0.5	63,875	66,449	61,876	57,694	61,514	61,162	62,784	65,978	59,107	64,852	62,529	2,701
GBHS Tagger con 0.3	63,875	66,449	61,876	57,694	61,514	61,162	62,784	65,978	59,107	64,852	62,529	**2,701**
GBHS Tagger con 0.0	65,702	67,059	61,803	58,681	62,034	62,323	64,311	67,039	60,017	66,387	63,536	2,842
HSTAGger 3	54,044	54,379	46,612	46,014	47,599	48,838	54,485	53,818	47,904	55,236	50,893	3,585
HSTAGger 2	60,423	62,941	54,627	53,181	56,040	55,993	60,187	61,453	53,488	61,239	57,957	3,468
HSTAGger	59,069	61,547	54,171	52,125	55,336	55,168	58,969	61,564	53,746	61,239	57,294	3,395
Azar	56,565	57,647	50,838	48,792	50,810	51,599	56,412	57,151	50,619	58,185	53,862	3,427

Tabla 5.11. Resultados de la ejecución de los algoritmos por folder.

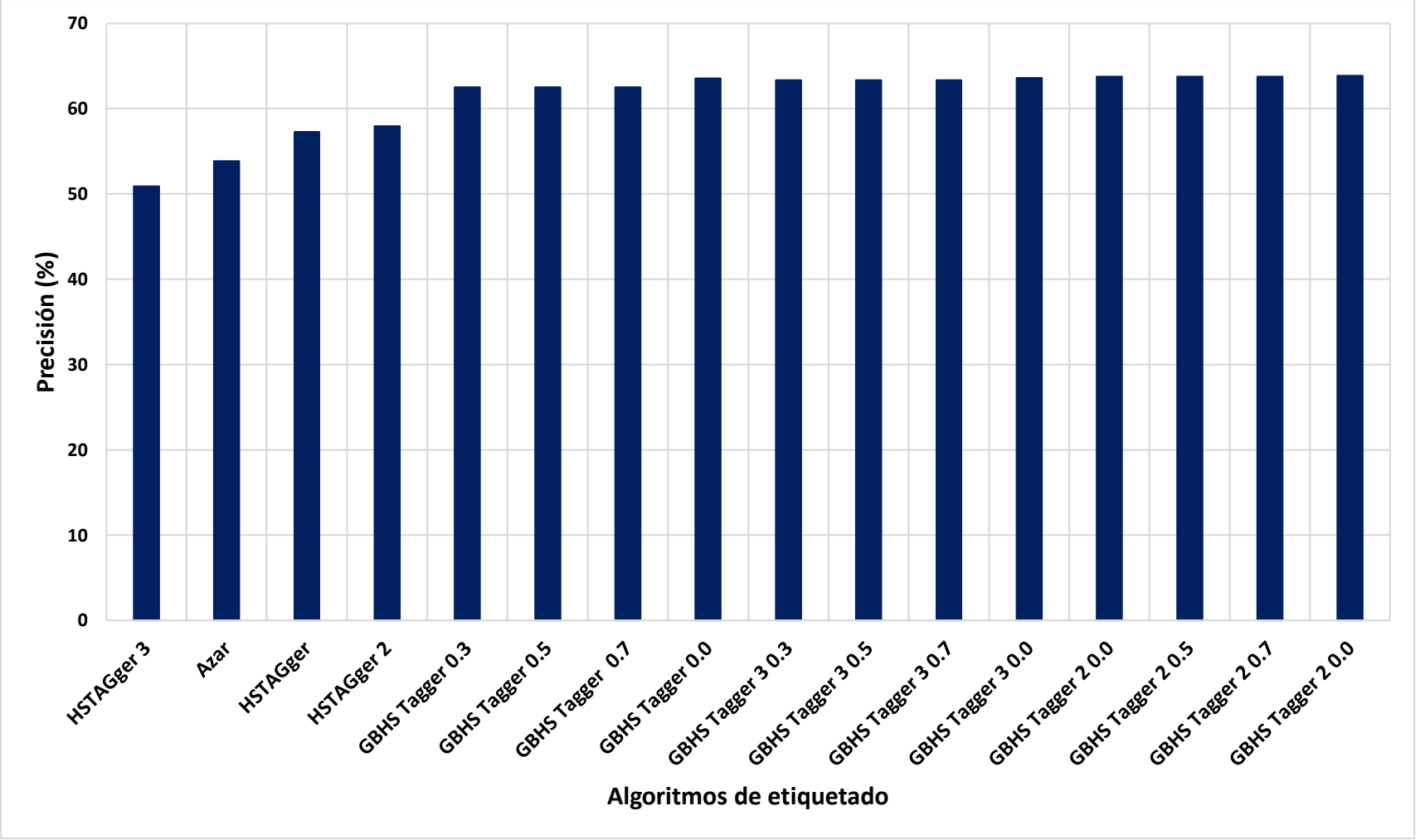

Figura 5.3. Comparación desempeño algoritmos de etiquetado para k= 10 folders.

En la **Figura 5.3**, se puede apreciar el desempeño de cada uno de los algoritmos evaluados. Se puede observar una diferencia significativa en los valores de precisión, confirmando que el algoritmo GBHS Tagger 2, se desempeña mejor, seguido por el algoritmo GBHS Tagger 3.

La **Tabla 5.11** muestra los valores de precision obtenido por cada folder, permitiendo apreciar que los resultados obtenidos por GBHS Tagger en todas sus versiones son mejores que los obtenidos por la línea base y las tres versiones de HSTAGger.

Al igual que con el experimento del corpus Brown, los Test de Friedman y de Wilcoxon fueron aplicados a estos datos, con el fin de establecer si las diferencias entre los algoritmos son estadísticamente significativas. La **Tabla 5.12**, muestra los resultados obtenidos al aplicar el test de Friedman (con p-value 5.756839449588824E-11), los cuales confirman que GBHS Tagger 2 sin optimización obtiene los mejores resultados, seguido por las versiones con optimización. Se puede apreciar que GBHS Tagger en sus diferentes versiones se desempeña mejor que los otros algoritmos.

Algoritmo	Ranking Friedman
GBHS Tagger 2 con 0.0	3.45
GBHS Tagger 2 con 0.3	4.55
GBHS Tagger 2 con 0.5	4.55
GBHS Tagger 2 con 0.7	4.55
GBHS Tagger 3 con 0.0	4.9
GBHS Tagger con 0.0	5.3
GBHS Tagger 3 con 0.3	7.3
GBHS Tagger 3 con 0.5	7.3
GBHS Tagger 3 con 0.7	7.3
GBHS Tagger con 0.3	9.6
GBHS Tagger con 0.5	9.6
GBHS Tagger con 0.7	9.6
HSTagger 2	13.25
HSTagger	13.75
Azar	15
HSTagger 3	16

Tabla 5.12. Ranking Test de Friedman para k=10 folders. **Fuente:** Propia resultados de KEEL [126]

El test de Wilcoxon muestra que:

- Con un 90% de confianza ➔ los resultados GBHS Tagger 2 sin optimizador local superan los algoritmos evaluados.
- Con un 95 % de confianza GBHS Tagger en todas sus versiones supera el desempeño de los etiquetadores de la línea base

5.2.4 Experimento 2 (Leave-one-out)

1. **Conjunto de datos.** El segundo experimento utilizó solo una oración como dato de prueba y las restantes 174 frases del corpus como datos de entrenamiento, es decir, que cuando se va a evaluar la oración 1, las oraciones correspondientes a los datos de entrenamiento son de la oración 2 a la 175, y así sucesivamente. La **Tabla 5.13**, muestra algunos ejemplos de los datos.

Oración de prueba	Oraciones en los datos de prueba	Palabras en los datos de prueba	Oraciones de entrenamiento	Palabras en los datos de entrenamiento	Palabras comunes	Palabras desconocidas
2	1	13	1,3 a 175	1989	8	5 (38.46 %)
5	1	24	1,2,3,4,6 a 175	1978	13	11 (45.83 %)
50	1	7	1 a 49, 51 a 175	1995	3	4 (57.14%)
80	1	34	1 a 79, 81 a 175	1968	20	14 (41.18%)
175	1	6	1 a 174	1996	1	5 (83.33%)

Tabla 5.13. Ejemplos de conjunto de datos de entrenamiento y prueba para Leave-One-Out. **Fuente:** Propia

Al igual que para el experimento 1, el porcentaje de palabras desconocidas es muy alto, lo cual se debe a que el corpus etiquetado para nasa yuwe es pequeño, afectando considerablemente el rendimiento de los algoritmos de etiquetado, no obstante, el desempeño mejora en relación con el experimento 1 que utiliza 10 folders. Lo anterior, se puede apreciar en los resultados presentados en la **Tabla 5.14**.

2. **Resultados**

La **Tabla 5.14** muestra los valores de precisión y desviación estándar de los algoritmos, donde los mejores resultados se pueden apreciar en el algoritmo GBHS Tagger, sin optimizador local. También se puede apreciar que el siguiente mejor resultado lo obtiene el mismo algoritmo, pero utilizando optimizador local. En todos los casos el mejor desempeño lo muestra GBHS Tagger en todas sus versiones, en contraste con los otros algoritmos evaluados.

Algoritmos	Parámetro (ProbOpt)	Número de oraciones	Precisión (%)	Desviación Estándar
Azar	-	175	57.7022	17.1942
HSTAGger	-	175	60.1914	17.0776
HSTAGger2	-	175	60.8964	17,3815
HSTAGger3	-	175	53.7983	16.6512
GBHS Tagger	**0.0**	**175**	**66.5787**	**16.9290**
GBHS Tagger	0.3	175	66.4297	17.6616
GBHS Tagger	0.5	175	66.4297	17.6616
GBHS Tagger	0.7	175	66.4297	17.6616
GBHS Tagger 2	0.0	175	65.9432	16.9991
GBHS Tagger 2	0.3	175	**66.2706**	17.4027
GBHS Tagger 2	0.5	175	66.2706	17.4027
GBHS Tagger 2	0.7	175	66.2706	17.4027
GBHS Tagger 3	0.0	175	65.9909	16.8131
GBHS Tagger 3	0.3	175	66.0765	17.4176
GBHS Tagger 3	0.5	175	66.0765	17.4176
GBHS Tagger 3	0.7	175	66.0765	17.4176

Tabla 5.14*.* Resultados de la ejecución de los algoritmos usando leave-one-out. **Fuente:** Propia

En la **Figura 5.4**, se puede apreciar, la comparación del comportamiento de los algoritmos evaluados .

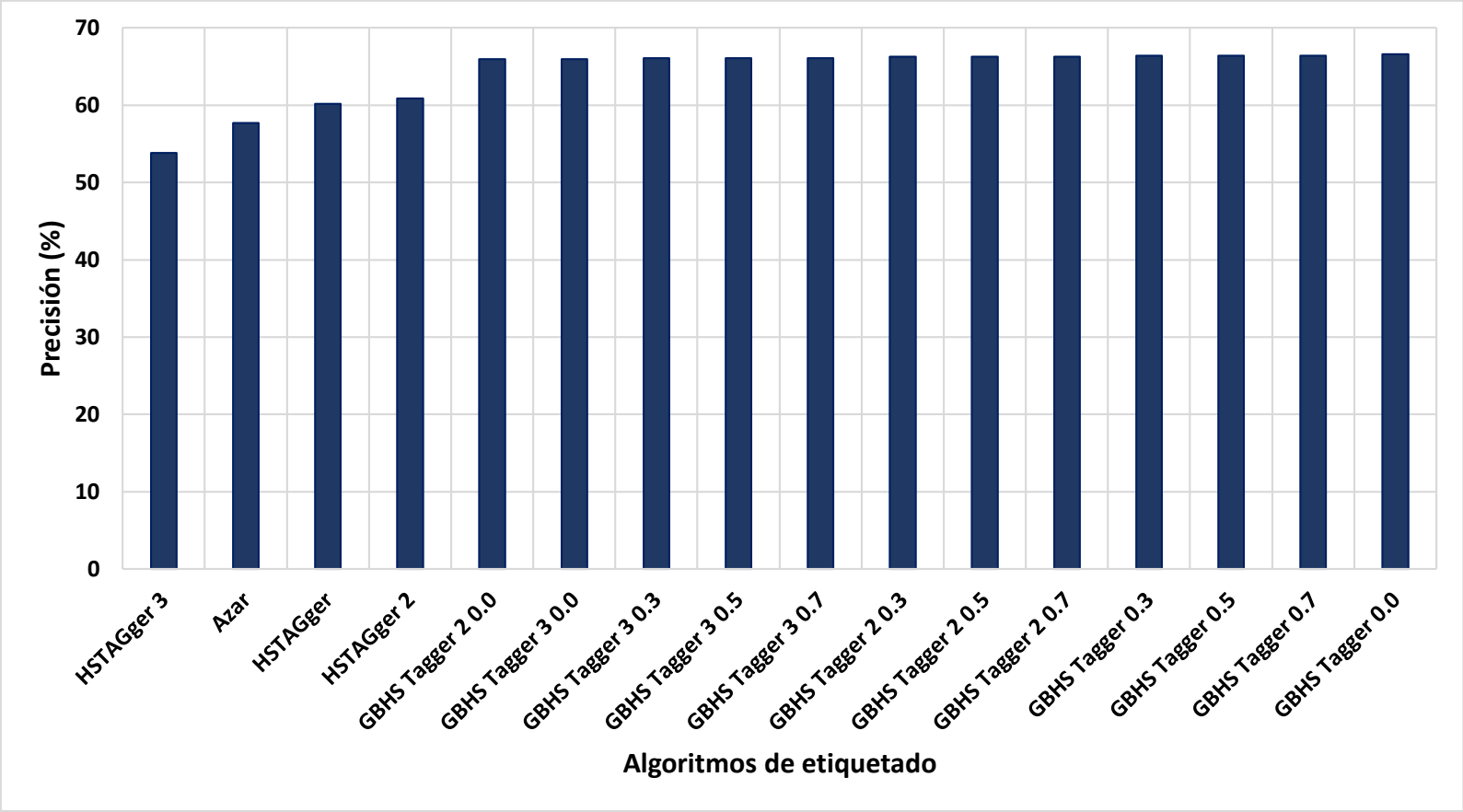

Figura 5.4. Comparación desempeño algoritmos de etiquetado para Leave-one-out. **Fuente:** Propia

La **Tabla 5.15**, muestra los resultados específicos de precisión para algunos folders (oraciones) presentados en la **Tabla 5.13**, en donde se puede apreciar que para todos los datos de prueba GBHS Tagger en todas sus versiones obtiene mejores resultados en relación a los otros algoritmos evaluados.

Algoritmos	Parámetro (ProbOpt)	Precisión Conjunto de prueba (oración de prueba)				Precisión palabras desconocidas			
		2	**5**	**80**	**175**	**2**	**5**	**80**	**175**
Azar	-	56,67	56,53	47,16	36,11	26.67	16.20	15.71	12.50
HSTAGger	-	61,03	58.19	46.67	45.56	40.00	22.70	14.88	36.00
HSTAGger2	-	59.49	63.06	49.78	45.00	24.00	18.18	19.66	35.29
HSTAGger3	-	54.87	50.97	42.84	32.22	11.66	12.50	9.11	22.00
GBHS Tagger	0.0	68.97	67.36	49.22	67.22	44.52	32.50	23.81	46.00
GBHS Tagger	0.3	71.03	62.08	50.10	**79.44**	47.33	33.80	24.60	76.92
GBHS Tagger	0.5	71.03	62.08	50.10	**79.44**	46.66	34.40	25.11	76.00
GBHS Tagger	0.7	71.03	62.08	50.10	**79.44**	46.21	34.09	25.37	76.00
GBHS Tagger 2	0.0	70.52	70.83	53.04	79.44	52.00	37.27	25.00	**77.77**
GBHS Tagger 2	0.3	72.56	71.11	52.94	73.89	50.32	37.45	22.46	72.00
GBHS Tagger 2	0.5	72.56	71.11	52.94	78.33	47.74	37.62	31.59	74.67
GBHS Tagger 2	0.7	72.56	71.11	52.94	78.33	50.67	37.27	31.59	73.79
GBHS Tagger 3	0.0	70.77	69.03	**51.47**	78.33	47.33	**39.39**	**37.09**	73.10
GBHS Tagger 3	0.3	**73.84**	**67.77**	52.45	78.89	52.67	33.94	22.49	71.11
GBHS Tagger 3	0.5	**73.84**	**67.77**	52.45	78.89	**53.33**	35.15	25.10	72.63
GBHS Tagger 3	0.7	**73.84**	**67.77**	52.45	78.89	52.63	35.15	34.42	69.17

Tabla 5.15. Algunos resultados por folder usando leave-one-out. **Fuente:** Propia

Los Test de Friedman y de Wilcoxon también fueron aplicados a estos datos, con el fin de establecer si existen diferencias estadísticamente significativas entre los algoritmos.

La **Tabla 5.16**, muestra los resultados obtenidos al aplicar el test de Friedman (p-value 2.662209341863786E-10), los cuales muestran que GBHS Tagger 2 con optimizador local se desempeña mejor, no obstante, también se puede concluir en todas las versiones de GBHS Tagger superan a los otros algoritmos evaluados.

Algoritmo	Ranking Friedman
GBHS Tagger 2 con 0.3	6.5857
GBHS Tagger 2 con 0.5	6.5857
GBHS Tagger 2 con 0.7	6.5857
GBHS Tagger 3 con 0.3	6.7086
GBHS Tagger 3 con 0.5	6.7086
GBHS Tagger 3 con 0.7	6.7086
GBHS Tagger 2 con 0.0	7.0457
GBHS Tagger 3 con 0.0	7.0657
GBHS Tagger con 0.0	7.3571
GBHS Tagger con 0.3	7.8371
GBHS Tagger con 0.5	7.8371
GBHS Tagger con 0.7	7.8371
HSTagger 2	11.3857
HSTagger	11.7171
Azar	13.3657
HSTagger 3	14.6686

Tabla 5.16*.* Ranking Test de Friedman para Leave-One-Out.

Adicionalmente, para este experimento el test de Wilcoxon mostró con una confianza del 90% que los resultados de GBHS Tagger supera los resultados de los otros algoritmos.

5.2.5 Medidas obtenidas de la matriz de confusión

Teniendo en cuenta el conjunto de etiquetas (clases) utilizado, es posible realizar una comparación entre ellas, la cual se ha realizado con las medidas (micro) de precisión, recuerdo y medida F, obtenidas de una matriz de confusión desarrollada para el mejor algoritmo de etiquetado. Los resultados obtenidos para cada uno de los experimentos se presentan a continuación. En la **Tabla 5.17** se presenta la matriz de confusión para el algoritmo gbhs Tagger 2 sin optimización local, para el experimento 1.

Etiqueta	Medidas Micro		
	Precisión (%)	Recuerdo (%)	Medida F (%)
Verb	66,6739	62,0286	64,2674
Noun	61,5784	65,3318	63,3996
Adj	50,4807	44,9813	47,5726
Adv	62,7291	68,9595	65,6969
Conj	58,5876	47,6596	52,5616
Det	46,4360	37,5522	41,5242
Num	2,0833	0,6623	1,0050
.	81,8224	100,0000	90,0026
Pron	22,5141	14,8331	17,8838
Promedio	62,9711	63,7643	63,1711

Tabla 5.17. Medidas Matriz de confusión para K= 10

En la **Tabla 5.18** se presenta la matriz de confusión para el algoritmo gbhs Tagger 2 con optimización local, el cual obtuvo el primer lugar según el Test Friedman, para el experimento 2.

Etiqueta	Medidas Micro		
	Precisión (%)	**Recuerdo (%)**	**Medida F (%)**
Verb	67,6725	62,2405	64,8429
Noun	64,6726	64,4131	64,5426
Adj	50,8625	46,8964	48,7990
Adv	63,8271	70,4409	66,9711
Conj	50,5311	47,2340	48,8270
Det	47,3641	47,8607	47,6111
Num	0,5435	0,6623	0,5970
.	80,9568	100,0000	89,4764
Pron	13,2911	15,7697	14,4247
Promedio	64,0941	64,2760	64,0329

Tabla 5.18. Medidas Matriz de confusión para Leave One Out

Como se mostró para el segundo experimento, los valores ponderados de micro precisión de la matriz de confusión son mejores, lo cual se presenta por que el tamaño de los datos de entrenamiento es un poco mayor.

En los valores ponderados de micro precisión para ambos experimentos, se puede apreciar que el etiquetado de signos de puntuación es alto, seguido por los verbos, adverbios, nombres, conjunciones y adjetivos. Mientras que para determinantes, pronombres y numerales son valores bajos.

Estos valores permiten tomar decisiones asociadas al enriquecimiento del corpus etiquetado de nasa yuwe, en relación con el contenido de las oraciones que se vayan a adicionar al corpus, por ejemplo, la inclusión de oraciones que contengan palabras con etiqueta Num y Pron, de tal forma, que se pueda contar con un mayor número de palabras con esta etiqueta y así mejorar los valores de desempeño del algoritmo etiquetador.

5.2.6 Síntesis

Los experimentos desarrollados en esta sección utilizan el corpus etiquetado para la lengua nasa yuwe, convirtiéndose en el primer trabajo de etiquetado para esta lengua. Se utilizaron varios algoritmos de etiquetado para cada experimento, como fueron: un algoritmo de etiquetado aleatorio, 3 versiones de HSTagger (descritas en la sección 5.1.3), y 3 versiones del algoritmo GBHS Tagger propuesto (descritas en el capitulo 4) cada una ejecutada con 4 valores diferentes para el parámetro ProbOpt (0.0, 0.3,0.5 y 0.7) que controla el uso del optimizador local.

El resultado de estos experimentos mostró que el algoritmo GBHS Tagger propuesto se desempeña mejor que los otros algoritmos evaluados, convirtiéndose en la mejor opción a la fecha para continuar esta línea de trabajo sobre la lengua nasa.

Los resultados obtenidos en los dos experimentos muestran significativas mejoras en los valores de desempeño de los 16 algoritmos de etiquetado en el experimento 2 con relación al experimento 1, como se puede apreciar en la **Figura 5.5**. Este incremento indica que el tamaño del corpus es relevante para el desempeño de los algoritmos. Lo que motiva a los grupos de investigación a ampliar el corpus etiquetado para Nasa Yuwe como parte de los futuros trabajos en el área.

Otro resultado importante de esta sección es un artículo titulado “Building a Nasa Yuwe Language Corpus and tagging with a metaheuristic approach”, el cual fue aprobado en la 19th International Conference on Computational Linguistics and Intelligent Text Processing. Conferencia que cuenta con publicaciones indexadas como por Colciencias. Este artículo presenta una breve descripción del proceso de construcción del corpus etiquetado y los experimentos mostrados en la **sección 5.2** de este documento. En el Anexo D, se encuentra el artículo enviado, que se encuentra pendiente de publicación.

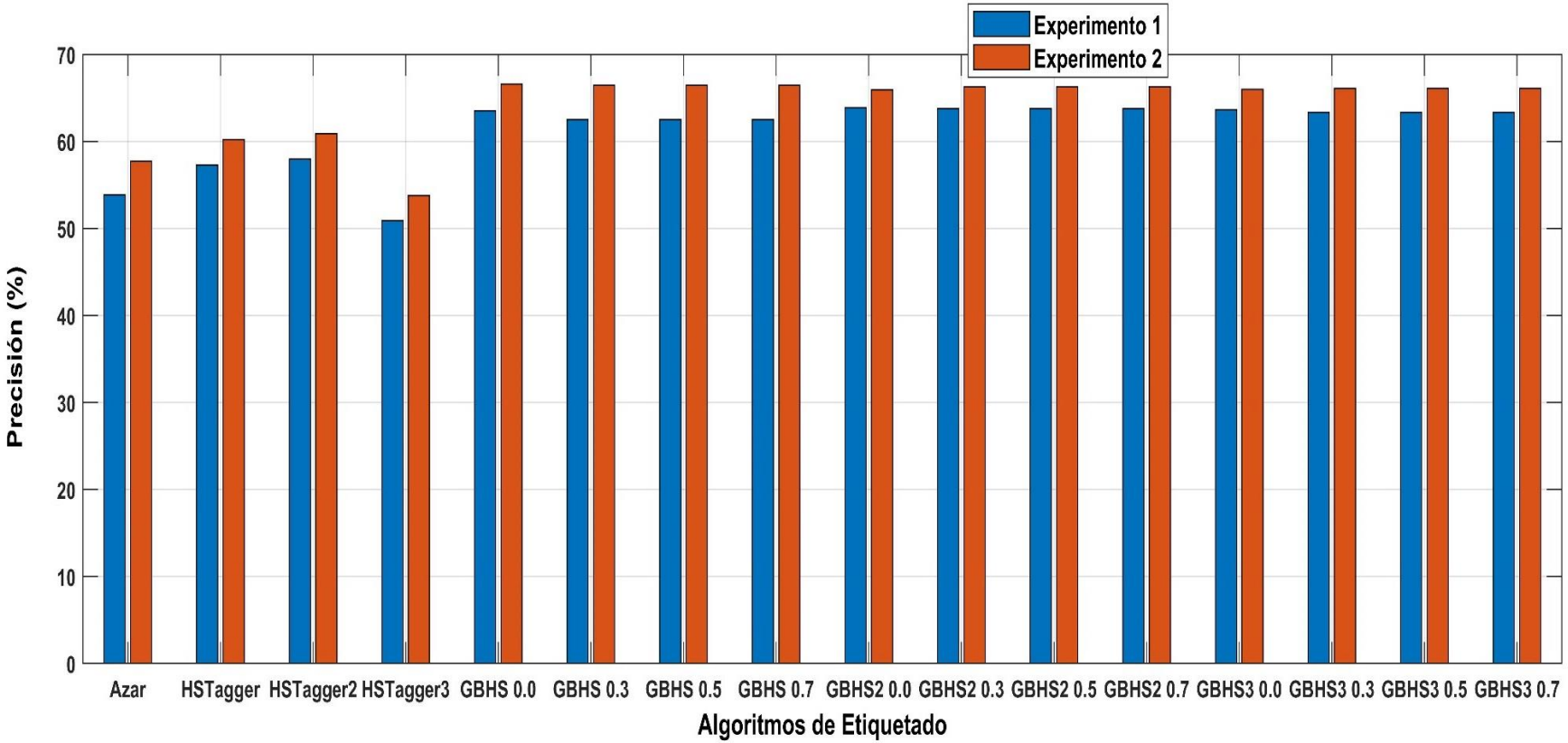

Figura 5.5. Comparación de los algoritmos de etiquetado en los experimentos sobre el corpus nasa. **Fuente:** Propia

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se presentan los comentarios finales relacionados con el desarrollo de este trabajo. En la sección 6.1, se describen las conclusiones y en la sección 6.2, el trabajo futuro.

6.1 Conclusiones

6.1.1 A nivel del proceso de etiquetado

El algoritmo GBHS Tagger en todas sus versiones es una propuesta para el problema de etiquetado desde la perspectiva de un problema de optimización, que encuentra la mejor secuencia de etiquetas para un conjunto de palabras en una oración, obteniendo mejores resultados en su desempeño, en comparación con los otros algoritmos evaluados. En donde, el mejor resultado fue obtenido por el algorimo propuesto, GBHS Tagger 2 para el caso de estudio del corpus de inglés, Precisión: 95.1959 y Precisión: 66.2706

La inclusión de conocimiento del problema en el algoritmo memético propuesto para el problema de etiquetado, a través de la utilización del optimizador local basado en la metaheurística subiendo la colina (Hill Climbing), demostró que este tipo de algoritmos son una alternativa eficiente para encontrar la mejor secuencia de etiquetado para una oración, al igual que lo han sido para las otras aplicaciones en las que se han utilizado.

El algoritmo de etiquetado basado en Memético propuesto en este trabajo, no incluyó:

- Pruebas con otros corpus de inglés u otras lenguas no tradicionales diferentes al nasa yuwe.
- Un proceso formal de afinamiento de los parámetros para cada uno de los casos de estudio.

6.1.2 A nivel de corpus etiquetado

Con el objetivo de enseñar y revitalizar un idioma en peligro de extinción (como es el caso de la lengua nasa o Paéz), se han propuesto varias soluciones computarizadas. Una de las soluciones más ampliamente recomendadas es desarrollar un corpus etiquetado que pueda ser utilizado en un proceso POST desde diferentes perspectivas (por ejemplo: traducción automática, etc.). En este sentido, el alcance de este trabajo puede expresarse en los siguientes resultados:

- En primer lugar, se lleva a cabo una descripción detallada del proceso de construcción de un corpus etiquetado, a través del análisis y la revisión de trabajos similares. Tal proceso implicó la definición del conjunto de etiquetas. El análisis presentado, en esta tesis, resalta las características de un lenguaje independiente como el caso del nasa yuwe, que aún se encuentra en proceso de descripción.

- En segundo lugar, se presenta un corpus etiquetado para nasa yuwe, el cual se encuentra alineado al conjunto de etiquetado universal, lo que favorece su posterior uso en diferentes aplicaciones. De tal forma, que este corpus constituye una contribución importante para el trabajo futuro con respecto a este lenguaje en particular, así como a otros idiomas que están en peligro de extinción y que no han sido materia de estudio para las investigaciones en procesamiento del lenguaje natural.

- En tercer lugar, este trabajo deja pendiente la consolidación y aplicación de una herramienta que pueda ser desplegada dentro dentro de una actividad de apoyo a procesos de enseñanza de la lengua.

6.1.3 A nivel de los experimentos realizados

Los experimentos realizados utilizaron validación cruzada que favorece la independencia entre los datos de entrenamiento y de prueba y permite llegar a una mejor estimación del desempeño de los algoritmos.

Cabe resaltar que en la realización de los experimentos presentados tanto para el algoritmo propuesto GBHS Tagger como para los algoritmos utilizados como línea base, se incluyeron en los datos de entrenamiento palabras conocidas como descoconocidas, obteniendo buenos resultados en el etiquetado de las palabras desconocidadas, una de las principales dificultades del problema de etiquetado.

Los resultados de precisión del algoritmo de etiquetado basado en meméticos propuesto, obtenidos en los experimentos realizados, muestran una diferencia entre el caso nasa yuwe y el caso de inglés, debido a que el corpus etiquetado nasa construido, es muy pequeño en relación al tamaño del corpus Brown, para el caso del inglés. Se pudo apreciar mejores valores de desempeño del algoritmo cuando se amplian los datos de entrenamiento en el experimento 2. Por tanto, esta tesis deja pendiente, evaluaciones del algoritmo propuesto, con otros corpus de lenguas no tradicionales y así como la ampliación del corpus etiquetado para nasa yuwe.

6.2 Trabajo futuro

El trabajo futuro se centrará en los siguientes aspectos clave:

- Enriquecer el corpus etiquetado para nasa yuwe tanto en su tamaño como en el conjunto de etiquetas utilizado, lo cual favorece el desempeño de los etiquetadores.

- Mejorar la función objetivo y la forma en que se hace el proceso de optimización local, dado que la inclusión del conocimiento potenciará aún más el desempeño del algoritmo.

- Desarrollar un proceso formal de afinamiento de los parámetros para el algoritmo propuesto.

- Proponer y evaluar otros algoritmos metaheurísticos para el problema de etiquetado tales como Evolución Diferencial [128] y Particle Swarm Optimization [58].

- Usar el algoritmo propuesto GBHS Tagger en sus diferentes versiones sobre otras lenguas tradicionales y no tradicionales, así como para otros corpus de inglés.

- Mejorar el etiquetador propuesto para la lengua nasa yuwe, con el objetivo de aumentar los valores de precisión. Para ello, se debe llevar a cabo el análisis de los diferentes métodos utilizados para crear un etiquetador (por ejemplo, técnicas estadísticas, entre otros) y la definición de una estrategia para identificar y asignar la etiqueta más probable para cada palabra en una oración.

- Hacer la construcción y despliegue de una herramienta informática de apoyo a procesos de enseñanza de la lengua nasa yuwe.

Bibliografía

[1] Moscato, P., & Cotta, C. (2010). Chapter 6 A Modern Introduction to Memetic Algorithms. In Handbook of Metaheuristics (pp. 105-144). Boston MA: Springer US. doi:10.1007/978-1-4419-1665-5

[2] Sean, L. (2012). Essentials of Metaheuristics (Online Version ed.). Lulu. Retrieved from http://cs.gmu.edu/~sean/book/metaheuristics/

[3] Brownlee, J. (2011). Clever Algorithms Nature-Inspired Programming Recipes. Melbourne: lulu.com.

[4] Yatsko, V. (2011). Methods and algorithms for automatic text analysis. Automatic Documentation and Mathematical Linguistics, 45(5), 224-231. doi:10.3103/S0005105511050062

[5] Klatt, S., & Bohnet, B. (2004). You Don't Have to Think Twice if You Carefully Tokenize. In Natural Language Processing – IJCNLP 2004 - Lecture Notes in Computer Science (Vol. 3248, pp. 299-309). Hainan Island, China: Springer Berlin Heidelberg. doi:10.1007/978-3-540-30211-7_32

[6] Lee, K., & Geem, Z. (2005). A new meta-heuristic algorithm for continuos engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering, 194, 3902-3933. doi:10.1016/j.cma.2004.09.007

[7] Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algoritm for solving optimization problems. In Applied Mathematics and Computation (pp. 1567-1579).

[8] Manning, C., Raghavan, P., & Shütze, H. (2009). An Introduction to Information Retrieval. Cambridge University Press.

[9] Sierra Martínez, L. M., Cobos Lozada, C. A., Corrales, J. C., & Rojas Curieux, T. (2015). Building a nasa yuwe Test Collection. In Computational Linguistics and Intelligent Text Processing Volume 9041 of the series Lecture Notes in Computer Science (pp. 112-123). El Cairo, Egipto: Springer International Publishing. doi:10.1007/978-3-319-18111-0_9

[10] Petrov, S., Das, D., & McDonald, R. (2012). A Universal Part-of-Speech Tagset. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC '12). Istambul.

[11] Xiao, R. (2010). Creation Corpus. In Handbook of Natural Language Processing (pp. 147 - 166). CRC Press. Retrieved 2016

[12] Vidal Esmorís, A. (2013). Tesis de Maestría: "Algoritmos Heurísticos en Optimización". Santiago de Compostela: Universidad Santiago de Compostela. Retrieved 2017, from http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_782.pdf

[13] Jurafsky, D., & Martin, J. H. (2017). Chapter 10. Part-of-Speech Tagging. In Speech and Language Processing. Draft of August 7, 2017.

[14] Linoff, G., & Berry, M. (2011). Data mining techniques: for marketing, sales, and customer relationship management (Third Edition ed.). Indianapolis, USA: John Wiley & Sons.

[15] NLTK Project. (2017, Septiembre). NLTK 3.2.5 documentation. Retrieved 2017, from http://www.nltk.org/

[16] Güngör, T. (2010). Part-of-Speech Tagging. In Handbook of Natural Language Processing (pp. 205-236). Boca Raton, FL: CRC Press.

[17] Pratt, K. S. (2009). Design Patterns for Research Methods: Iterative Field Research. Retrieved Octubre 2013, from http://www.kpratt.net/wp-content/uploads/2009/01/research_methods.pdf

[18] Silva, A. P., Silva, A., & Rodríguez, I. (2013). PSO-Tagger: A New Biologically Inspired Approach to the Part-of-Speech Tagging Problem. In Proceedings of ICANNGA 2013, LNCS 7824 (pp. 90-99). Lausanne, Switzerland: Springer-Verlag Berlin Heidelberg.

[19] Forsati, R., & Shamsfard, M. (2015). Novel harmony search-based algorithms for part-of-speech tagging. Knowledge and Information Systems, 42(3), 709-736.

[20] Spiegel, M. R, & Stephens, L. J.(2009). Estadística, México D.F.: McGrawHIll.

[21] Zumel, N., & Mount, J. (2014). Practiccal Data Science with R. Shelter Island, NY: Manning Publications.

[22] Lantz, B. (2013). Machine Learning with R. Birmingham-Mumbai: Packt Publishing.

[23] Corte Constitucional Consejo Superior de la Judicatura. CONSTITUCIÓN POLÍTICA DE COLOMBIA 1991 (2016). [En línea]. Available: http://www.corteconstitucional.gov.co/inicio/Constitucion%20politica%20de%20Colombia.pdf. [Último acceso: 2018].

[24] Attia, M., Rashwan, M., & Al-Badrashiny, M. (2009, July). Fassieh (R), a Semi-Automatic Visual Interactive Tool for Morphological, PoS-Tags, Phonetic, and Semantic Annotation of Arabic Text Corpora. IEEE Transactions on Audio, Speech, and Language Processing, 17(5), 916 - 925.

[25] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. New York: Pearson - Addison Wesley.

[26] Silva, A., Silva, A., & Rodrigues, I. (2012). An Approach to the POS Tagging Problem Using Genetic Algorithms. In Springer, Computational Intelligence Volume 577 of the series Studies in Computational Intelligence (pp. 3-17). Spain: Springer International Publishing.

[27] Paul, A., Purkayastha, B., & Sarkar, S. (2015). Hidden Markov Model Based Part of Speech Tagging for Nepali Language. 2015 International Symposium on

Advanced Computing and Communication (ISACC) (pp. 149 - 156). Silchar: IEEE. doi:10.1109/ISACC.2015.7377332

[28] Brants, T. (2000). TnT - a statistical part-of-speech tagger. Proceedings of the sixth conference on Applied natural language processing ANLC '00 (pp. 224-231). Stroudsburg, PA, USA: Association for Computational Linguistics.

[29] Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-of-Speech Tagging. Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 133-142). Association for Computational Linguistics.

[30] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International Conference on Machine Learning 2001 (pp. 282-289). ACM.

[31] Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics, 21(4), 543–565.

[32] Schmid, H. (1994). Part-of-speech tagging with neural networks. Proceedings of the 15th conference on computational linguistics. 1, pp. 172-176. Stroudsburg, PA, USA: Association for Computational Linguistics.

[33] Makazhanov, A., Yessenbayev, Z., Sabyrgaliyev, I., & Sharafudinov, A. (2014). On Certain Aspects of Kazakh Part-of-Speech. 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), (pp. 1-4). Astana: IEEE. doi:10.1109/ICAICT.2014.7035953

[34] Das, D., & Petrov, S. (2011). Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (pp. 600-609). Stroudsburg, PA, USA: Association for Computational Linguistics.

[35] Carneiro, H., França, F. M., & Lima, P. M. (2015). Multilingual part-of-speech tagging with weightless neural networks. Neural Networks, 66, 11-21.

[36] Ariaratnam, I., Weerasinghe, A., & Liyanage, C. (2014). A shallow parser for Tamil. 2014 International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 197 - 203). Colombo: IEEE. doi:10.1109/ICTER.2014.7083901

[37] Duong, L., Cook, P., Bird, S., & Pecina, P. (2013). Simpler unsupervised POS tagging with bilingual projections. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013 - Volume 2: Short Papers). (pp. 634-639). Sofia, Bulgaria: Association for Computational Linguistics.

[38] Amazings Divulgación S.L. (n.d.). La navaja de Ockham para explicar el creacionismo. (Amazings Divulgación S.L) Retrieved from http://naukas.com/2011/01/28/la-navaja-de-ockham-para-explicar-el-creacionismo/

[39] Araujo, L. (2007). How evolutionary algorithms are applied to statistical natural language processing. Artificial Intelligence Review, 28(4), 275-303.

[40] Alba, E., Luque, G., & Araujo, L. (2006). Natural language tagging with genetic algorithms. Information Processing Letters, 100(5), 173-182.

[41] Forsati, R., & Shamsfard, M. (2012). Cooperation of Evolutionary and Statistical statistical PoS-tagging. 2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP) (pp. 446 - 451). Shiraz, Fars: IEEE.

[42] Ekbal, A., & Saha, S. (2013). Simulated annealing based classifier ensemble techniques: Application to part of speech tagging. Information Fusion, 14(3), 288-300.

[43] Silva, A., Silva, A., & Rodrigues, I. (2013). A New Approach to the POS Tagging Problem Using Evolutionary. Proceedings of Recent Advances in Natural Language Processing (pp. 619–625). Hissar, Bulgaria: Elservier.

[44] Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. IEEE Transactions on evolutionary computation, 1(1), 67-82.

[45] Aggarwal, C. C., Orlin, J. B., & Tai, R. P. (1997). Optimized crossover for the independent set problem. Operations Research, 45(2), 226-236.

[46] Berretta, R., Cotta, C., & Moscato, P. (2003). Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm: Results on the number partitioning problem. In Metaheuristics: Computer-Decision Making Volume 86 of the series Applied Optimization (pp. 65-90). Springer US.

[47] Merz, P., & Freisleben, B. (2000). Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. Evolutionary Computation, 8, 61-91.

[48] Merz, P. (2002). A comparison of memetic recombination operators for the traveling salesman problem. GECCO 2002: Proceedings of the Genetic and Evolutionary (pp. 472-479). New York: Morgan Kaufmann Publishers.

[49] Rojas, T. E. (2012). Esbozo Gramatical de la lengua nasa (lengua Paéz). In UNICEF (Ed.), El Lenguaje en Colombia. Tomo I: Realidad Lingüístidca de Colombia. Bogotá: Academia Colombiana de la Lengua e Instituto Caro y Cuervo.

[50] Daelemans, W. (2010). Encyclopedia of Machine Learning (Part of Speech Tagging). (C. Sammut, & G. Webb, Eds.) Springer US.

[51] Nogueira dos Santos, C., & Luiz Milidiú, R. (2012). Part-of-Speech Tagging. In Entropy Guided Transformation Learning: Algorithms and Applications SpringerBriefs in Computer Science (pp. 35-41). Springer London.

[52] Rezende Fernandes, E., Muller Rodrigues, I., & Luiz Milidiú, R. (2014). Portuguese Part-of-Speech Tagging with Large Margin Structure Learning. 2014 Brazilian Conference on Intelligent Systems (BRACIS), (pp. 25-30). Sao Paulo: IEEE.

[53] Paul, A., Purkayastha, B. S. & Sarkar, S. I. (2015). Hidden Markov Model Based Part of Speech Tagging for Nepali Language de 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar.

[54] Wang, X., Zhang, J., & Yan, Y. (2010). Support Vector Machine for Chinese Part-Of-Speech Tagging in Speech Synthesis Systems. Proceedings of 2010

International Conference on Biomedical Engineering and Computer Science. Wuhan: IEEE.

[55] Mayfield, J., McNamee, P., Piatko, C., & Pearce, C. (2003). Lattice-based Tagging using Support Vector Machines. Proceeding CIKM '03 Proceedings of the twelfth international conference on Information and knowledge management. New Orleans. doi:10.1145/956863.956921

[56] Yang, X.-S. (2009). Harmony Search as a Metaheuristic Algorithm. In Music-Inspired Harmony Search Algorithm Studies in Computational Intelligence (Vol. 191, pp. 1-14). Springer Berlin Heidelberg.

[57] Omran, M., & Mahdavi, M. (2008). Global-best harmony search. Applied Mathematics and Computation, 198, 643-656.

[58] Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theroy. Proceedings of the Sixth International Symposium on Micromachine and Human Science, (pp. 39-43).

[59] Glover, F., & Laguna, M. (1997). Tabu Search. MA, USA: Kluwer Academic Publisher. Retrieved from https://dl.acm.org/citation.cfm?id=549765

[60] Melian Batista, B., & Glover, F. (2007). Introduction to Tabu Search. In E. Crespo, R. Marti, & J. Pacheco (Ed.), Metaheuristic Procedures in Economics and Business Enterprise, (pp. 29-71). Retrieved from http://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS_Spanish%20w%20Belen%2811-9-06%29.pdf

[61] Neri, F., & Cotta, C. (2012). A Primer on Memetic Algorithms. In F. Neri, C. Cotta, & P. Moscato (Eds.), Handbook ofMemeticAlgorithms (pp. 43-52). Springer-Verlag Berlin Heidelberg.

[62] Cotta, C. (2007). Una Visión General de los Algoritmos Meméticos. Rect@: Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA, 3, 139-166.

[63] Hao, J.-K. (2012). Memetic Algorithms in Discrete Optimization. In Handbook of Memetic Algorithms (pp. 73-94). Springer-Verlag Berlin Heidelberg.

[64] Reeves, C. (1996). Hybrid genetic algorithms for bin-packing and related problems. Annals of Operations Research. 63, pp. 1996., 63(3), 371-396.

[65] Instituto Colombiano de Cultura Hispánica. (2008). Geografía Humana de Colombia. Región Andina Central (Vol. Tomo IV Volumen II). Bogotá. Retrieved Abril 4, 2008, from http://www.banrep.gov.co/blaavirtual/geografia/geohum2/indice.htm.

[66] Rojas Curieux, T. (2006). Por los caminos de la recuperación de la lengua Paéz (nasa yuwe). Popayán: Letrarte editores.

[67] Rojas, T. (2005). En la reflexión sobre lo oral y lo escrito: Educación escolar y práctica en pueblos indígenas. Editorial Universidad del Cauca.

[68] Rivet, P. (1913). Les familles linguistiques du Nord-Ouest de l'Amérique du Sud" en Année Linguistique (Société Philologique). Journal de la Société des Américanistes. n°1, 1913., Tome 10(1), 117-154.

[69] Greenberg, J. (1987). Language in the Americas. California: Stanford University Press. Retrieved from http://www.sup.org/books/title/?id=2693

[70] Loukotka, C. (1968). Classification of South American Indian Languages. Los Angeles: Latin American Studies Center, University of California.

[71] Constenla, A. (1993). "La Familia Chibcha". In Estado Actual de la Clasificación de las Lenguas Indígenas de Colombia (pp. 75-125). Santafé de Bogotá: Imprenta Patriótica del Instituto Caro y Curevo.

[72] Landaburu, J. (2000). Clasificación de las lenguas indígenas de Colombia. In Lenguas Indígenas de Colombia: una visión descriptiva (pp. 25-48). Santafé de Bogotá: Imprenta Patriótica del Instituto Caro y Cuervo.

[73] Jung, I. (1984). Gramática del Páez o nasa yuwe. Descripción de una Lengua Indígena de Colombia. Published by LINOM GmbH 2008.

[74] CRIC y el Programa de Dllo Rural en la Región de Tierra Dentro Cxhab Wala - PT/CW. (2005). Diccionario Nasa Yuwe - Castellano (Primera ed.). Popayán: Litografía San José.

[75] Rojas, T. (1998). La Lengua páez. Bogotá: Ministerio de Cultura.

[76] Rojas C., T., Perdomo Dizú, A., & Corrales Carvaja, M. H. (2009). Una Mirada al nasa yuwe de Novirao (Primera ed.). Popayán: Sello Editorial Universidad del Cauca.

[77] Platt, J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. Advances in Large Margin Classifiers (pp. 61-74). MIT Press.

[78] Ekbal, A., & Bandyopadhyay, S. (2008). Part of Speech Tagging in Bengali Using Support Vector Machine. International Conference on Information Technology, 2008. ICIT'08 (pp. 106 - 111). IEEE. doi:10.1109/ICIT.2008.12

[79] Ismail, S., Rahman, M., & Al Mumin, M. (2014). Developing an Automated Bangla Parts of Speech. 16th International Conference on Computer and Information Technology (ICCIT) (pp. 355 - 359). Khulna: IEEE. doi:10.1109/ICCITechn.2014.6997347

[80] Kardan, A. A., & Bahojb Imani, M. (2014). Improving Persian POS tagging using the maximum entropy model. 2014 Iranian Conference on Intelligent Systems (ICIS). Bam, Iran: IEEE. doi:10.1109/IranianCIS.2014.6802567

[81] Raja, F., Amiri, H., Tasharofi, S., Sarmadi, M., Hojjat, H., & Oroumchian, F. (2007). Evaluation of Part of Speech Tagging of Persian Text. Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages.

[82] Albared, M., Al-Moslmi, T., Omar, N., Al-Shabi, A., & Ba-Alwi, F. M. (2016). Probabilistic Arabic Part of Speech Tagger with Unknown Words Handling. Journal of Theoretical and Applied Information Technology, 90(2), 236-246. Retrieved from http://www.jatit.org/volumes/Vol90No2/26Vol90No2.pdf

[83] Samuelsson, C. (1996). Handling sparse data by successive abstraction. Proceedings of the 16th conference on Computational linguistics, 2, pp. 895–900. Copenhagen, Denmark. doi:10.3115/993268.993323

[84] Keyaki, A., & Miyazaki, J. (2017). Part-of-speech tagging for web search queries using a large-scale web corpus. Proceedings of the Symposium on Applied Computing, (pp. 931-937). Marrakech, Morocco. doi:10.1145/3019612.3019694

[85] Brill, E. (1992). A simple rule-based part of speech tagger. Proceedings of the third conference on Applied natural language processing (ANLC '92) (pp. 152-155). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/974499.974526

[86] AlSuhaibani, R., Newman, C., Collard, M., & Maletic, J. (2015). Heuristic-Based Part-of-Speech Tagging of Source Code Identifiers and Comments. 2015 IEEE 5th Workshop on Mining Unstructured Data (MUD) (pp. 1-6). Bremen: IEEE. doi:10.1109/MUD.2015.7327960

[87] Aziz T, A., & Sunitha, C. (2015). A Hybrid Parts of Speech Tagger for Malayalam. 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1502 - 1507). Kochi: IEEE. doi:10.1109/ICACCI.2015.7275825

[88] Nakamura, M., & Shikano, K. (1989). A study of English word category prediction based on neutral networks, Acoustics, Speech, and Signal Processing. International Conference on Acoustics, Speech, and Signal Processing IEEE. 2, pp. 731-734. Glasgow: IEEE. doi:10.1109/ICASSP.1989.266531

[89] Pérez-Ortiz, J., & Forcada, M. L. (2001). Part-of-Speech Tagging with recurrent neural networks. Proceedings of International Joint Conference on Neural

Networks, IJCNN '01. (pp. 1588 - 1592 vol.3). IJCNN '01.: IEEE. doi:10.1109/IJCNN.2001.938396

[90] Kabir, F., Abdullah-Al-Mamun, K., & Nurul Huda, M. (2016). Deep learning-based parts of speech tagger for Bengali. 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016. Dhaka, Bangladesh: IEEE. doi:10.1109/ICIEV.2016.7760098

[91] Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Choudhury, M., Nath Jha, G. KVS Subbarao. (2014). A Common Parts-of-Speech Tagset Framework for Indian Languages. Proceedings of LREC 2008, (pp. 1331-1337). Marrakech. Retrieved from http://www.lrec-conf.org/proceedings/lrec2008/pdf/337_paper.pdf

[92] Hnin, H., Pa Pa, W., & Thu, Y. (2017). Back-Propagation Neural Network Approach to Myanmar Part-of-Speech Tagging. In Advances in Intelligent Systems and Computing (pp. 212-220). Springer, Cham. doi:10.1007/978-3-319-48490-7_25

[93] Zennaki, O., Semmar, N., & Besacier, L. (2015). Unsupervised and Lightly Supervised Part-of-Speech Tagging Using Recurrent Neural Networks. 29th Pacific Asian Conference on Language, Information and Computation, (pp. 133-142). Shangai, China.

[94] Duong, L., Cohn, T., Verspoor, K., Bird, S., & Cook, P. (2014). What a Can We Get From 1000 Tokens? A Case Study of Multilingual POS Tagging for Resource-Poor Languages. Proceedings of Conference on Empirical Methods in Natural Language Processing (pp. 886-897). Doha, Qatar: Association for Computational Linguistics.

[95] Lv, C., Liu, H., & Dong, Y. (2010). An Efficient Corpus Based Part-of-Speech Tagging with GEP. 2010 Sixth International Conference on Semantics Knowledge and Grid (SKG). Beijing, China: IEEE. doi:10.1109/SKG.2010.42

[96] Fang, K., & Ma, C. (1994). Orthogonal and Uniform Design for Experiment. Beijing: Science Press.

[97] Lv, C., Liu, H., Dong, Y., Li, F., & Liang, Y. (2017). Using Uniform-Design GEP for Part-of-Speech Tagging. Journal of Circuits, Systems and Computers, 26(4), 1-14. doi:doi.org/10.1142/S0218126617500608

[98] Forsati, R., Shamsfard, M., & Mojtahedpour, P. (2010). An Efficient Meta Heuristic Algorithm for POS-Tagging. 2010 Fifth International Multi-Conference on Computing in the Global Information Technology (ICCGI) (pp. 93- 98). Valencia: IEEE.

[99] Forsati, R., & Shamsfard, M. (2014). Hybrid PoS-tagging: A cooperation of evolutionary and statistical approaches. Applied Mathematical Modelling, 38(13), 3193-3211.

[100] Silva, A. P., Silva, A., & Rodrigues, I. (2012). Tagging with Disambiguation Rules A New Evolutionary Approach to the Part-of-Speech Tagging Problem. Proceedings of the 4th International Joint Conference on Computational Intelligence (ECTA-2012) (pp. 5-14). SCITEPRESS (Science and Technology Publications, Lda.).

[101] Silva, A. P., Silva, A., & Rodríguez, I. (2014). Part-of-Speech Tagging Using Evolutionary Computation. In Nature Inspired Cooperative Strategies for Optimization (NICSO 2013) Volume 512 of the series Studies in Computational Intelligence (pp. 167-178). Springer International Publishing. doi:10.1007/978-3-319-01692-4_13

[102] Bachir Menai, M. E. (2014). Word sense disambiguation using evolutionary algorithms – Application to Arabic language. Computers in Human Behavior, 41, 92-103.

[103] Ekbal, A., & Saha, S. (2011). A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in Indian languages as case studies. Expert Systems with Applications, 38, 14760–14772.

[104] Lourdes, A. (2002). Part-of-Speech Tagging with Evolutionary Algorithms. CICLing 2002, LNCS 2276, 2002, (pp. 230–239).

[105] Wilson, G., & Heywood, M. (2005). Use of a Genetic Algorithm in Brill's Transformation-Based Part-of-Speech Tagger. Proceedings of GECCO'05, Genetic and Evolutionary Computation Conference. Washington, DC, USA: ACM.

[106] Rabbi, I., Abid Khan, M., & Ali, R. (2008). Developing a Tagset for Pashto Part of Speech Tagging. Second International Conference on Electrical Engineering. Lahore (Pakistan). doi:10.1109/ICEE.2008.4553909

[107] Expert Advisory Group on Language Engineering Standards. (1996, Mar). EAGLES Recommendations for the Morphosyntatic Annotation of Corpora. Retrieved Enero 11, 2017, from http://home.uni-leipzig.de/burr/Verb/htm/LinkedDocuments/annotate.pdf

[108] Francis, W., & Kucera, H. (1979, July). Brown Corpus. Retrieved from Brown University: http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM#bc8

[109] Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: the penn treebank. Journal Computational Linguistics - Special issue on using large corpora: II, 19(2), 313-330.

[110] Dinakaramani, A., Rashel, F., Luthfi, A., & Manurung, R. (2014). Designing an Indonesian Part of speech Tagset and Manually Tagged Indonesian Corpus. 2014 International Conference on Asian Language Processing (IALP) (pp. 66 - 69). Kuching: IEEE.

[111] Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. Proceedings of the Tenth Machine Translation Summit (MT Summit XX) (pp. 79-86). Phuket, Thailand: AAMT.

[112] Ahmed Mahar, J., & Qadir Memon, G. (2010). Rule Based Part of Speech Tagging of Shindi Language. 2010 International Conference on Signal Acquisition and Processing (pp. 101-106). Washington, DC, USA: IEEE Computer Society Washington, DC, USA. doi:10.1109/ICSAP.2010.27

[113] Spoustová, J., & Spousta, M. (2012). A High-Quality Web Corpus of Czech. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). Istanbul, Turkey}. Retrieved from http://www.lrec-conf.org/proceedings/lrec2012/summaries/120.html

[114] Singh, S., & Banerjee, E. (2014). Annotating Bhojpuri Corpus using BIS Scheme. 2014 Proceedings of 2nd Workshop on Indian Language Data: Resources and Evaluation (WILDRE-2), Ninth International Conference on Language Resources and Evaluation, LREC 2014. Reykjavik, Iceland. Retrieved from http://www.lrec-conf.org/proceedings/lrec2014/index.html

[115] Scherrer, Y., Nerima, L., Russo, L., Ivanova, M., & Wehrli, E. (2014). SwissAdmin: A multilingual tagged parallel corpus of press releases. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014). Reykjavik, Iceland. Retrieved from http://archive-ouverte.unige.ch/unige:38811

[116] Bach, N. X., Linh, N. D. y Phuong, T. M. (2018). An Empirical Study on POS Tagging for Vietnamese Social Media Text. Computer Speech and Language, vol. 50, pp. 1-15.

[117] Jianchao, T. (2015). An English Part of Speech Tagging Method Based on Maximum Entropy. 2015 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), (pp. 76-80). Halong Bay, Vietnam: IEEE. doi:10.1109/ICITBS.2015.25

[118] Ranjan Das, B., Sahoo, S., Sekhar Panda, C., & Patnaik, S. (2015). Part of Speech tagging in odia using support vector machine. Procedia Computer Science. International Conference on Intelligent Computing, Communication Converge (ICCC-2015), 48, 507-512. doi: 10.1016/j.procs.2015.04.127

[119] Zhonglin, Y., Zhen, J., Huang, J., & Hongfeng, Y. (2016). Part-of-Speech Tagging based on Dictionary and Statistical Machine Learning. Proceedings of the 35th Chinese Control Conference (CCC). Chengdu, China: IEEE. doi:10.1109/ChiCC.2016.7554459

[120] Sun, W., & Wan, X. (2016). Towards Accurate and Efficient Chinese Part-of-Speech Tagging. Computational Linguistics, 42(3), 391-419. doi:10.1162/COLI a 00253

[121] Mall, S., & Jaiswal, U. (2015). Innovative Algorithms for Parts of Speech Tagging in Hindi-English Machine. Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on (pp. 709 - 714). Noida: IEEE.

[122] Tian, Y., & Lo, D. (2015). A Comparative Study on the Effectiveness of Part-of-Speech Tagging Techniques on Bug Reports. 2015 IEEE 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 570 -574). Montreal, QB: IEEE. doi:10.1109/SANER.2015.7081879

[123] Poel, M., Boschman, E., & Akker, R. (2008). A Neural Network Based Dutch Part of Speech Tagger. Proceedings of the twentieth Belgian-Dutch Artificial Intelligence Conference (BNAIC 2008), (pp. 217-224). The Netherlands.

[124] Carneiro, H. d., França, F., & Lima, P. M. (2010). WANN-Tagger - A Weightless Artificial Neural Network Tagger for the Portuguese Language. Proceedings of the International Conference on Fuzzy Computation and International Conference on Neural Computation ICFC-ICNC 2010 (pp. 330-335). Valencia: SciTePress.

[125] Jurafsky, D., & Martin, J. (2010). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education.

[126] Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M., Ventura, S., Garrell, J., Herrera, F. (2009). KEEL: a software tool to assess evolutionary algorithms. Soft Comput, 13(3), 307-318. doi: 10.1007/s00500-008-0323-y

[127] Sierra Martínez, L. M., Cobos, C., & Corrales, J. C. (2017). Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging. In A. Ghosh, R. Pal, & R. Prasath (Ed.), The Fifth International Conference on Mining Intelligence and Knowledge Exploration. Lecture Notes in

Computer Science, vol 10682, pp. 198-211. IDRBT, Hyderabad, India: Springer, Cham. doi: https://doi.org/10.1007/978-3-319-71928-3_20

[128] Sierra, L. M., Cobos, C., & Corrales, J. C. (2014). Continuous Optimization Based on a Hybridization of Differential Evolution with K-means. In Advances In: Artificial Intelligence Iberamia 2014 Lecture Notes In Computer Science (pp. 381 - 392). Santiago de Chile: Springer International Publishing. doi:10.1007/978-3-319-12027-0_31

[129] Rojas Curieux, T. E. (1998). La lengua paéz una visión de su gramática (primera ed.). (M. d. Cultura, Ed.) Bogotá, Colombia: Panamericana Formas e Impresos S.A.

Anexo B.

Detalles Corpus Etiquetado Brown

¶

B.1. Conjunto de etiquetas Brown

Tag	Description	Examples
.	sentence closer	. ; ? !
(	left paren	
)	right paren	
*	*not, n't*	
--	dash	
,	comma	
:	colon	
ABL	pre-qualifier	*quite, rather*
ABN	pre-quantifier	*half, all*
ABX	pre-quantifier	*both*
AP	post-determiner	*many, several, next*
AT	article	*a, the, no*
BE	*be*	
BED	*were*	
BEDZ	*was*	
BEG	*being*	
BEM	*am*	
BEN	*been*	
BER	*are, art*	
BEZ	*is*	
CC	coordinating conjunction	*and, or*
CD	cardinal numeral	*one, two, 2, etc.*
CS	subordinating conjunction	*if, although*
DO	*do*	
DOD	*did*	
DOZ	*does*	
DT	singular determiner	*this, that*
DTI	singular or plural determiner/quantifier	*some, any*
DTS	plural determiner	*these, those*
DTX	determiner/double conjunction	*either*
EX	existentil *there*	
FW	foreign word (hyphenated before regular tag)	
HL	word occurring in headline (hyphenated after regular tag)	
HV	*have*	

Tag	Description	Examples
HVD	*had* (past tense)	
HVG	*having*	
HVN	*had* (past participle)	
HVZ	*has*	
IN	preposition	
JJ	adjective	
JJR	comparative adjective	
JJS	semantically superlative adjective	*chief, top*
JJT	morphologically superlative adjective	*biggest*
MD	modal auxiliary	*can, should, will*
NC	cited word (hyphenated after regular tag)	
NN	singular or mass noun	
NN$	possessive singular noun	
NNS	plural noun	
NNS$	possessive plural noun	
NP	proper noun or part of name phrase	
NP$	possessive proper noun	
NPS	plural proper noun	
NPS$	possessive plural proper noun	
NR	adverbial noun	*home, today, west*
NRS	plural adverbial noun	
OD	ordinal numeral	*first, 2nd*
PN	nominal pronoun	*everybody, nothing*
PN$	possessive nominal pronoun	
PP$	possessive personal pronoun	*my, our*
PP$$	second (nominal) possessive pronoun	*mine, ours*
PPL	singular reflexive/intensive personal pronoun	*myself*
PPLS	plural reflexive/intensive personal pronoun	*ourselves*
PPO	objective personal pronoun	*me, him, it, them*
PPS	3rd. singular nominative pronoun	*he, she, it, one*
PPSS	other nominative personal pronoun	*I, we, they, you*
QL	qualifier	*very, fairly*
QLP	post-qualifier	*enough, indeed*
RB	adverb	
RBR	comparative adverb	
RBT	superlative adverb	
RN	nominal adverb	*here then, indoors*
RP	adverb/particle	*about, off, up*
TL	word occurring in title (hyphenated after regular tag)	
TO	infinitive marker *to*	
UH	interjection, exclamation	
VB	verb, base form	
VBD	verb, past tense	
VBG	verb, present participle/gerund	

Tag	Description	Examples
VBN	verb, past participle	
VBZ	verb, 3rd. singular present	
WDT	*wh*- determiner	*what, which*
WP$	possessive *wh*- pronoun	*whose*
WPO	objective *wh*- pronoun	*whom, which, that*
WPS	nominative *wh*- pronoun	*who, which, that*
WQL	*wh*- qualifier	*how*
WRB	*wh*- adverb	*how, where, when*

B.2. Equivalencias conjunto de etiqueta Brown con conjunto de etiquetado universal

ETIQUETA BROWN	ETIQUETA PETROV
‘	.
(	.
(-hl	.
)	.
)-hl	.
*	adv
*-hl	adv
*-nc	adv
*-tl	adv
,	.
,-hl	.
,-nc	.
,-tl	.
--	.
.	.
.-hl	.
.-nc	.
.-tl	.
:	.
:-hl	.
:-tl	.
abl	prt
abn	prt
abn-hl	prt
abn-nc	prt
abn-tl	prt
ben	verb
ben-tl	verb

ETIQUETA BROWN	ETIQUETA PETROV
abx	det
ap	adj
ap$	prt
ap+ap-nc	adj
ap-hl	adj
ap-nc	adj
ap-tl	adj
at	det
at-hl	det
at-nc	det
at-tl	det
at-tl-hl	det
be	verb
be-hl	verb
be-tl	verb
bed	verb
bed*	verb
bed-nc	verb
bedz	verb
bedz*	verb
bedz-hl	verb
bedz-nc	verb
beg	verb
bem	verb
bem*	verb
bem-nc	verb
dt+bez	prt
dt+bez-nc	prt

ETIQUETA BROWN	ETIQUETA PETROV
ber	verb
ber*	verb
ber*-nc	verb
ber-hl	verb
ber-nc	verb
ber-tl	verb
bez	verb
bez*	verb
bez-hl	verb
bez-nc	verb
bez-tl	verb
cc	conj
cc-hl	conj
cc-nc	conj
cc-tl	conj
cc-tl-hl	conj
cd	num
cd$	noun
cd-hl	num
cd-nc	num
cd-tl	num
cd-tl-hl	num
cs	adp
cs-hl	adp
cs-nc	adp
cs-tl	adp
do	verb
do*	verb
do*-hl	verb
do+ppss	x
do-hl	verb
do-nc	verb
do-tl	verb
dod	verb
dod*	verb
dod*-tl	verb
dod-nc	verb
doz	verb
doz*	verb
doz*-tl	verb
doz-hl	verb
doz-tl	verb
dt	det
dt$	det

ETIQUETA BROWN	ETIQUETA PETROV
dt+md	prt
dt-hl	det
dt-nc	det
dt-tl	det
dti	det
dti-hl	det
dti-tl	det
dts	det
dts+bez	prt
dts-hl	det
dtx	det
ex	prt
ex+bez	prt
ex+hvd	prt
ex+hvz	prt
ex+md	prt
ex-hl	prt
ex-nc	prt
fw-*	x
fw-*-tl	x
fw-at	x
fw-at+nn-tl	x
fw-at+np-tl	x
fw-at-hl	x
fw-at-tl	x
fw-be	x
fw-ber	x
fw-bez	x
fw-cc	x
fw-cc-tl	x
fw-cd	x
fw-cd-tl	x
fw-cs	x
fw-dt	x
fw-dt+bez	x
fw-dts	x
fw-hv	x
fw-in	x
fw-in+at	x
fw-in+at-t	x
fw-in+at-tl	x
fw-in+nn	x
fw-in+nn-tl	x
fw-in+np-tl	x

ETIQUETA BROWN	ETIQUETA PETROV
fw-in-tl	x
fw-jj	x
fw-jj-nc	x
fw-jj-tl	x
fw-jjr	x
fw-jjt	x
fw-nn	x
fw-nn$	x
fw-nn$-tl	x
fw-nn-nc	x
fw-nn-tl	x
fw-nn-tl-nc	x
fw-nns	x
fw-nns-nc	x
fw-nns-tl	x
fw-np	x
fw-np-tl	x
fw-nps	x
fw-nps-tl	x
fw-nr	x
fw-nr-tl	x
fw-od-nc	x
fw-od-tl	x
fw-pn	x
fw-pp$	x
fw-pp$-nc	x
fw-pp$-tl	x
fw-ppl	x
fw-ppl+vbz	x
fw-ppo	x
fw-ppo+in	x
fw-pps	x
fw-ppss	x
fw-ppss+hv	x
fw-ql	x
fw-rb	x
fw-rb+cc	x
fw-rb-tl	x
fw-to+vb	x
fw-uh	x

ETIQUETA BROWN	ETIQUETA PETROV
fw-vb	x
fw-vb-nc	x
fw-vb-tl	x
fw-vbd	x
fw-vbd-tl	x
fw-vbg	x
fw-vbg-tl	x
fw-vbn	x
fw-vbz	x
fw-wdt	x
fw-wpo	x
fw-wps	x
hv	verb
hv*	verb
hv+to	verb
hv-hl	verb
hv-nc	verb
hv-tl	verb
hvd	verb
hvd*	verb
hvd-hl	verb
hvg	verb
hvg-hl	verb
hvn	verb
hvz	verb
hvz*	verb
hvz-nc	verb
hvz-tl	verb
in	adp
in+in	adp
in+ppo	adp
in-hl	adp
in-nc	adp
in-tl	adp
in-tl-hl	adp
jj	adj
jj$-tl	prt
jj+jj-nc	adj
jj-hl	adj
jj-nc	adj

ETIQUETA BROWN	ETIQUETA PETROV
fw-uh-nc	x
fw-uh-tl	x
jj-tl-nc	adj
jjr	adj
jjr+cs	adj
jjr-hl	adj
jjr-nc	adj
jjr-tl	adj
jjs	adj
jjs-hl	adj
jjs-tl	adj
jjt	adj
jjt-hl	adj
jjt-nc	adj
jjt-tl	adj
md	verb
md*	verb
md*-hl	verb
md+hv	verb
md+ppss	verb
md+to	verb
md-hl	verb
md-nc	verb
md-tl	verb
nil	x
nn	noun
nn$	noun
nn$-hl	noun
nn$-tl	noun
nn+bez	prt
nn+bez-tlprt	
nn+hvd-tl	prt
nn+hvz	prt
nn+hvz-tl	prt
nn+in	noun
nn+md	prt
nn+nn-nc	noun
nn-hl	noun
nn-nc	noun
nn-tl	noun

ETIQUETA BROWN	ETIQUETA PETROV
jj-tl	adj
jj-tl-hl	adj
nns$-hl	noun
nns$-nc	noun
nns$-tl	noun
nns$-tl-hl	noun
nns+md	prt
nns-hl	noun
nns-nc	noun
nns-tl	noun
nns-tl-hl	noun
nns-tl-nc	noun
np	noun
np$	noun
np$-hl	noun
np$-tl	noun
np+bez	prt
np+bez-nc	prt
np+hvz	prt
np+hvz-nc	prt
np+md	prt
np-hl	noun
np-nc	noun
np-tl	noun
np-tl-hl	noun
nps	noun
nps$	noun
nps$-hl	noun
nps$-tl	noun
nps-hl	noun
nps-nc	noun
nps-tl	noun
nr	noun
nr$	noun
nr$-tl	noun
nr+md	prt
nr-hl	noun
nr-tl	noun
nr-tl-hl	noun
nr-nc	noun

ETIQUETA BROWN	ETIQUETA PETROV
nn-tl-hl	noun
nn-tl-nc	noun
nns	noun
nns$	noun
od-nc	adj
od-tl	adj
pn	noun
pn$	noun
pn+bez	prt
pn+hvd	prt
pn+hvz	prt
pn+md	prt
pn-hl	noun
pn-nc	noun
pn-tl	noun
pp$	det
pp$$	pron
pp$-hl	det
pp$-nc	det
pp$-tl	det
ppl	pron
ppl-hl	pron
ppl-nc	pron
ppl-tl	pron
ppls	pron
ppo	pron
ppo-hl	pron
ppo-nc	pron
ppo-tl	pron
pps	pron
pps+bez	prt
pps+bez-hl	prt
pps+bez-nc	prt
pps+hvd	prt
pps+hvz	prt
pps+md	prt
pps-hl	pron
pps-nc	pron
pps-tl	pron
ppss	pron

ETIQUETA BROWN	ETIQUETA PETROV
nrs	noun
nrs-tl	noun
od	adj
od-hl	adj
ppss+bez*	prt
ppss+hv	prt
ppss+hv-tl	prt
ppss+hvd	prt
ppss+md	prt
ppss+md-nc	prt
ppss+vb	prt
ppss-hl	pron
ppss-nc	pron
ppss-tl	pron
ql	adv
ql-hl	adv
ql-nc	adv
ql-tl	adv
qlp	adv
rb	adv
rb$	prt
rb+bez	prt
rb+bez-hl	prt
rb+bez-nc	prt
rb+cs	adv
rb-hl	adv
rb-nc	adv
rb-tl	adv
rbr	adv
rbr+cs	adv
rbr-nc	adv
rbt	adv
rn	adv
rp	prt
rp+in	prt
rp-hl	prt
rp-nc	prt
rp-tl	prt
to	prt
to+vb	prt

ETIQUETA BROWN	ETIQUETA PETROV
ppss+bem	prt
ppss+ber	prt
ppss+ber-n	prt
ppss+ber-nc	prt
ppss+ber-tl	prt
ppss+bez	prt
uh-tl	prt
vb	verb
vb+at	verb
vb+in	verb
vb+jj-nc	verb
vb+ppo	verb
vb+rp	verb
vb+to	verb
vb+vb-nc	verb
vb-hl	verb
vb-nc	verb
vb-tl	verb
vbd	verb
vbd-hl	verb
vbd-nc	verb
vbd-tl	verb
vbg	verb
vbg+to	verb
vbg-hl	verb
vbg-nc	verb
vbg-tl	verb
vbn	verb
vbn+to	verb
vbn-hl	verb
vbn-nc	verb
vbn-tl	verb
vbn-tl-hl	verb
vbn-tl-nc	verb
vbz	verb
vbz-hl	verb
vbz-nc	verb
vbz-tl	verb
wdt	det
wdt+ber	prt

ETIQUETA BROWN	ETIQUETA PETROV
to-hl	prt
to-nc	prt
to-tl	prt
uh	prt
uh-hl	prt
uh-nc	prt
wdt-hl	det
wdt-nc	det
wp$	det
wpo	pron
wpo-nc	pron
wpo-tl	pron
wps	pron
wps+bez	prt
wps+bez-nc	prt
wps+bez-tl	prt
wps+hvd	prt
wps+hvz	prt
wps+md	prt
wps-hl	pron
wps-nc	pron
wps-tl	pron
wql	adv
wql-tl	adv
wrb	adv
wrb+ber	prt
wrb+bez	prt
wrb+bez-tl	prt
wrb+do	prt
wrb+dod	prt
wrb+dod*	prt
wrb+doz	prt
wrb+in	prt
wrb+md	prt
wrb-hl	adv
wrb-nc	adv
wrb-tl	adv
“	.
“	.
wdt+bez-tl	prt

ETIQUETA BROWN	ETIQUETA PETROV
wdt+ber+pp	x
wdt+bez	prt
wdt+bez-hl	prt
wdt+bez-nc	prt

ETIQUETA BROWN	ETIQUETA PETROV
wdt+do+pps	x
wdt+dod	prt
wdt+hvz	prt