**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**


# APRENDIZAJE AUTOMÁTICO NO SUPERVISADO EN SEGMENTADORES MORFOLÓGICOS PARA UNA LENGUA DE ESCASOS RECURSOS

CASO DE ESTUDIO: SHIWILU

**Tesis para obtener el título profesional de Ingeniera Informática**

**AUTORA:**

Evelyn Fiorella Asmat Ramirez

**ASESOR:**

Mag. Claudia Maria Del Pilar Zapata Del Río

Mag. Félix Arturo Oncevay Marcos

Lima, mayo, 2023

**Informe de Similitud**

Yo, Claudia María del Pilar Zapata Del Río, docente de la Facultad de Ciencias e Ingeniería de la Pontificia

Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado Aprendizaje automático no supervisado en segmentadores morfológicos para una lengua de escasos recursos. Caso de estudio: Shiwilu, del/de la autor(a)/ de los(as) autores(as) Evelyn Fiorella Asmat Ramírez,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 13%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 01/07/2022.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima 24 de marzo 2023

| | |
|---|---|
| Apellidos y nombres del asesor / de la asesora:<br>Zapata Del Río, Claudia María del Pilar | |
| DNI: 10799864 | Firma  |
| ORCID: 0000-0003-0134-6487 | |

**RESUMEN**

El Shiwilu es considerada ‘seriamente en peligro’ porque es hablada principalmente por adultos mayores de forma parcial, poco frecuente y en contextos restringidos; además, no continúa siendo transmitida a nuevas generaciones. Este tipo de lenguas necesitan pasar por un proceso de revitalización (fortalecimiento) para garantizar que no se extingan y así fomentar el interés de sus hablantes. Además, su documentación es muy escasa debido a los pocos estudios lingüísticos realizados. A fin de elevar su status, se sugiere la creación de recursos y tecnología de corte lingüístico, como corpus monolingüe y bilingüe, diccionarios, reconocimiento de categorías gramaticales, analizadores morfológicos, etc. Sin embargo, la mayoría de las lenguas existentes no se beneficia con alguno de estos recursos y/o tecnologías, y por ello son consideradas como lenguas de escasos recursos. Debido a la falta de inversión, se requiere un enfoque en el que se busquen soluciones robustas a un bajo costo a través de herramientas independientes de la lengua, modelos de desarrollo de código abierto o algoritmos de aprendizaje automático no supervisado. Bajo este contexto, se identifica como problema central el desconocimiento de un enfoque adecuado para la segmentación morfológica de una lengua de escasos recursos; y para ello, el presente proyecto propone realizar una segmentación morfológica automática no supervisada en una lengua con estas características a partir de la identificación del tipo de enfoque, monolingüe o multilingüe, que ofrece mejores resultados en esta tarea.

**DEDICATORIA**

A Dios que, con su infinito amor, supo guiar mis pasos.

A mi abuela, que está en el cielo, por los valores y principios inculcados.

A mi madre que, gracias a su gran esfuerzo, promovió cada uno de mis sueños y me ha permitido culminar mis estudios universitarios.

A mis tíos Jorge y Carlos por su cariño y apoyo constante durante toda mi vida.

A mi primo Eduardo por todos los momentos de felicidad, para que este sea un ejemplo y motivación en su futura vida universitaria.

**AGRADECIMIENTOS**

A mi esposo Carlos por su gran amor,

paciencia y confianza.

A mi padre y mi hermana Cecilia por sus

consejos en esta última etapa.

A mi asesor Arturo Oncevay por guiarme con

éxito en este proyecto.

A mi co-asesora Claudia Zapata por su apoyo

durante mi carrera universitaria.

**Tabla de Contenido**

## Índice de Figuras

**Índice de Tablas**

# Capítulo 1. Generalidades

## 1.1 Problemática

De acuerdo a la última edición de Ethnologue (2019), existen 7111 lenguas, de las cuales, 2895 se encuentran en peligro de extinción -poco más del 40% de las lenguas existentes a nivel mundial-, conformadas por menos de 1000 hablantes. En muchos casos, desaparecen junto al último hablante y sin dejar registro alguno. Desde el punto de vista de estos hablantes, existen lenguas consideradas de mayor prestigio, lo cual genera un patrón de preferencia de los padres en la enseñanza de sus hijos -omitiendo su lengua materna-, con el fin de mejorar sus oportunidades en la macrosociedad en que se encuentran (Winter, 1993).

Por lo tanto, el estatus de una lengua es una característica percibida, y como tal, puede verse afectado por la acción humana. A fin de elevar este estatus, se sugiere la creación de recursos y tecnología de corte lingüístico (Borin, 2009), como corpus monolingüe y bilingüe, diccionarios, reconocimiento de categorías gramaticales, analizadores morfológicos, etc. (Scannell, 2007). Sin embargo, la mayoría de las lenguas existentes no se beneficia con alguno de estos recursos y/o tecnologías, y por ello son consideradas como lenguas de escasos recursos (Scannell, 2007).

Por otra parte, debido a la falta de inversión, se requiere un enfoque en el que se busquen soluciones robustas a un bajo costo a través de herramientas independientes de la lengua, modelos de desarrollo de código abierto o algoritmos de aprendizaje automático no supervisado (Scannell, 2007). Por ejemplo, los métodos no supervisados permiten que se generen, de forma rápida y económica, recursos básicos de tecnología lingüística para nuevas lenguas (Hammarström & Borin, 2011).

El aplicar aprendizaje automático para extraer patrones de una lengua puede tratarse a diferentes niveles, por ejemplo, a nivel morfológico (Bender, 2013). La morfología es una de

las subdisciplinas más antiguas de la lingüística que estudia la variación sistemática de la formación y significado de las palabras, es decir, de su estructura interna (Hammarström & Borin, 2011).

El estudio morfológico tiene como objetivo describir y explicar los patrones de las unidades más pequeñas, llamadas morfemas, en los lenguajes humanos (Haspelmath, 2010). El aprendizaje automático no supervisado aplicado a la morfología permite describir cómo se construyen las palabras a partir de texto plano. Este análisis morfológico es un componente básico en otras aplicaciones de procesamiento de lenguaje natural como reconocimiento de voz o traductores automáticos (Hammarström & Borin, 2011).

Además, los enfoques del aprendizaje automático proveen a los lingüistas una ayuda computacional para el trabajo de documentación lingüística, evitando un extenso análisis manual que permita descubrir los componentes morfológicos de una lengua, especialmente para aquellas lenguas cuyos sistemas morfológicos involucran miles de formas flexionadas posibles para una sola palabra (Hammarström & Borin, 2011).

En las lenguas aislantes o analíticas, se podrían tener palabras con un solo morfema; y en otros casos, en las lenguas sintéticas, una palabra puede contener múltiples afijos, es decir, morfemas antes y después de la raíz (Bender, 2013).

Generalmente, los afijos ya tienen una ubicación determinada dentro de una palabra, es decir, puede ser un prefijo, sufijo, infijo o circunfijo, pero la cantidad por cada tipo de afijo es incierta (Bender, 2013). En la **Tabla 1**, tomando como ejemplo una lengua de abundantes recursos (español) se aprecia que podríamos tener palabras solo con sufijos (en el ejemplo “aguaceros” tiene 3 sufijos) o palabras con prefijos y sufijos (en el ejemplo “atemorizado” tiene 1 prefijo y 4 sufijos). Por lo tanto, se tiene palabras con cantidad variable de prefijos y/o sufijos por palabra.

**Tabla 1** Ejemplo de variación de número de tipo de morfemas por palabra en la segmentación morfológica. Adaptado de Sanchez (2014)

| **Aguaceros**<br>**(sustantivo plural)** | **Atemorizado**<br>**(adjetivo calificativo masculino singular)** |
|---|---|
| *agu–* (lexema)<br>*–ac–* (sufijo nominal)<br>*–ero–* (sufijo denominal)<br>*–s* (morfema flexivo nominal de número plural) | *a–* (prefijo)<br>*–temor–* (lexema)<br>*–iz–* (sufijo verbal)<br>*–a–* (morfema flexivo verbal: vocal temática)<br>*–d–* (morfema de participio pasado)<br>*–o* (morfema flexivo nominal de género masculino)<br>*Ø* (morfema flexivo nominal de número singular, morfo cero) |

Otra dimensión en la que se diferencian las lenguas es en el grado de complejidad de la identificación de la posición de inicio y fin del morfema. Para las lenguas aglutinantes, en las cuales varios afijos pueden ser añadidos a la raíz de una palabra, modificando su significado o función gramatical (Richards & Schmidt, 2013), estos límites son más claros, mientras que existen otros casos donde es bastante complejo identificarlos debido a procesos fonológicos sofisticados, falta de adecuación a secuencias de fonos o cambios en la raíz de la palabra (Bender, 2013). Tenemos en la **Tabla 2** un ejemplo en lengua quechua y aimara – lenguas aglutinantes. Se puede apreciar cómo se realiza una segmentación morfológica, donde cada elemento tiene su propio significado (Cerrón-Palomino, 1994).

**Tabla 2** Ejemplo de segmentación morfológica en lenguas aglutinantes. Adaptado de Cerrón-Palomino (1994)

| **Quechua** | **Aimara** | **Significado en español** |
|---|---|---|
| wasi | uta | ‘casa’ |
| wasi-kuna | uta-naka | ‘casas’ |
| wasi-kuna-wan | uta-naka-mpi | ‘con las casas’ |
| wasi-kuna-wan-mi | uta-naka-mpi-wa | ‘con las casas, ciertamente’ |

Sin embargo, según Haspelmath (2010), una de las complicaciones más frecuentes es que los morfemas pueden tener diferentes formas fonológicas bajo diferentes circunstancias, llamados alomorfos. Por ejemplo, en el español, se puede utilizar el morfema “s” o el morfema “es” para formar el plural.

Es decir, dada una lengua, a simple vista no podría definirse con certitud la cantidad de morfemas posibles ni la ubicación en la palabra; teniendo mayor dificultad las lenguas pobremente descritas o sin descripción alguna, siendo el caso de la mayoría de las lenguas (Crowley, 2007). De esta forma, el proceso de segmentación e identificación de los morfemas de una palabra es complejo para un lingüista, debido a la presencia de alomorfos, y por tratarse de un proceso de conjetura y verificación por la falta de conocimiento en la lengua a pesar de tener conocimientos en otras (Samarin, 1967).

Por otra parte, es difícil recolectar datos representativos de la formación de palabras de la mayoría de las lenguas, debido a que los estudios realizados se enfocan generalmente en lenguas más conocidas (Štekauer, Valera & Kőrtvélyessy, 2012). Sin embargo, la reciente expansión de la diversidad lingüística en los textos digitales ha hecho posible que el procesamiento de lenguaje natural se oriente cada vez más hacia el multilingüismo; teniendo como mayor reto la escasez de recursos (O'Horan, Berzak, Vulić, Reichart & Korhonen, 2016). Por ello, se han elaborado recursos de manera manual como grandes conjuntos de datos anotados lingüísticamente (bancos de árboles o corpus paralelos); y copiosas bases de datos de léxicos (terminologías o diccionarios) (O'Horan et al., 2016).

Estos recursos clave están disponibles para lenguas bien investigadas (por ejemplo, inglés, alemán y chino), pero para la mayoría de las otras lenguas faltan por completo. Además, la creación de estos recursos es costosa y por eso no puede realizarse para todas las lenguas existentes (O'Horan et al., 2016). Sin embargo, dado que varias lenguas de escasos recursos comparten algunas características con lenguas de abundantes recursos, podría

reducirse la cantidad de datos necesaria. Además, en muchos casos, las lenguas de escasos recursos son hablados en entornos multilingües donde existe alguna lengua de abundantes recursos, haciendo que los prestamos lingüísticos entre estas lenguas sean comunes. Por lo tanto, estos recursos permiten utilizarse a través de un enfoque multilingüe (Baumann & Pierrehumbert, 2014).

Bajo este contexto, se identifica como problema central el desconocimiento de un enfoque adecuado para la segmentación morfológica de una lengua de escasos recursos; y para ello, el presente proyecto de fin de carrera propone realizar una segmentación morfológica automática no supervisada en una lengua con estas características a partir de la identificación del tipo de enfoque, monolingüe o multilingüe, que ofrece mejores resultados en esta tarea.

En el Perú, existen 4 lenguas ‘en peligro’ y 17 lenguas ‘seriamente en peligro’ (Acosta, Natalia, Huamancayo, Mori, & Carbajal, 2013). El Shiwilu es considerada ‘seriamente en peligro’ porque es hablada principalmente por adultos mayores de forma parcial, poco frecuente y en contextos restringidos; además, no continúa siendo transmitida a nuevas generaciones (Acosta et al., 2013).

Este tipo de lenguas necesitan pasar por un proceso de revitalización (fortalecimiento) para garantizar que no se extingan y así fomentar el interés de sus hablantes (Acosta et al., 2013). Además, su documentación es muy escasa debido a los pocos estudios lingüísticos realizados (Valenzuela, 2008). Por ello, se tomará como caso de estudio esta lengua, y se planea publicar en la web los recursos usados (diccionarios y textos), además de un segmentador morfológico, para apoyar en la revitalización y difusión de este lenguaje en serio peligro de extinción.

## 1.2 Objetivos

### 1.2.1 Objetivo general

Realizar una segmentación morfológica automática no supervisada en una lengua de escasos recursos identificando la idoneidad del tipo de enfoque (monolingüe o multilingüe) que ofrece mejores resultados, tomando como caso de estudio la lengua shiwilu.

### 1.2.2 Objetivos específicos

**O1.** Procesar los recursos en shiwilu para el entrenamiento de modelos monolingües y multilingües que segmenten morfológicamente de manera automática no supervisada.

**O2.** Comparar segmentadores morfológicos automáticos no supervisados sobre una lengua de escasos recursos a través de un enfoque monolingüe

**O3.** Implementar un segmentador morfológico para una lengua de escasos recursos a través de un enfoque multilingüe junto a otra lengua de abundantes recursos

**O4.** Comparar morfemas extraídos de enfoques monolingüe y multilingüe para una lengua de escasos recursos

### 1.2.3 Resultados esperados

**R1.** Software que realiza la extracción de las palabras en shiwilu (O1)

**R2.** Listado de palabras shiwilu en su forma base y flexionada (O1)

**R3.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Morfessor (O2)

**R4.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Linguistica (O2)

**R5.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado al método de Byte Pair Encoding (O2)

**R6.** Segmentador morfológico automático a través de enfoque multilingüe haciendo uso de recursos en shiwilu e inglés (O3)

**R7.** Experimentación numérica para la selección del enfoque más preciso para segmentar morfológicamente la lengua shiwilu (O4)

### 1.2.4 Mapeo de objetivos, resultados y verificación

**Tabla 3** Resultados, meta física y medios de verificación para el objetivo específico 1

| **Objetivo 1:** Procesar los recursos en shiwilu para el entrenamiento de modelos monolingües y multilingües que segmenten morfológicamente de manera automática no supervisada | | |
|---|---|---|
| **Resultados** | **Meta física** | **Medio de verificación** |
| **R1.** Software que realiza la extracción de las palabras en shiwilu | Programa de extracción de palabras. | Palabras válidas en diccionario shiwilu (13 377) |
| **R2.** Listado de palabras shiwilu en su forma base y flexionada | Palabras en shiwilu | |

**Tabla 4** Resultados, meta física y medios de verificación para el objetivo específico 2

| **Objetivo 2:** Comparar segmentadores morfológicos automáticos no supervisados sobre una lengua de escasos recursos a través de un enfoque monolingüe | | |
|---|---|---|
| **Resultados** | **Meta física** | **Medio de verificación** |
| **R3.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Morfessor<br><br>**R4.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Linguistica<br><br>**R5.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado al método de Byte Pair Encoding | Comparación estadística de segmentadores morfológicos | Los morfemas obtenidos deben pertenecer realmente a la lengua. |

**Tabla 5** Resultados, meta física y medios de verificación para el objetivo específico 3

| **Objetivo 3:** Implementar un segmentador morfológico para una lengua de escasos recursos a través de un enfoque multilingüe junto a otra lengua de abundantes recursos | | |
|---|---|---|
| **Resultados** | **Meta física** | **Medio de verificación** |
| **R6.** Segmentador morfológico automático a través de enfoque multilingüe haciendo uso de recursos en shiwilu e inglés | Modelo algorítmico con enfoque multilingüe | Los morfemas obtenidos deben pertenecer realmente a la lengua. |

**Tabla 6** Resultados, meta física y medios de verificación para el objetivo específico 4

| **Objetivo 4:** Comparar morfemas extraídos de enfoques monolingüe y multilingüe para una lengua de escasos recursos | | |
|---|---|---|
| **Resultados** | **Meta física** | **Medio de verificación** |
| **R7.** Selección del enfoque más preciso para segmentar morfológicamente la lengua shiwilu | Comparación entre enfoques | Segmentador morfológico publicado en la web para la verificación de expertos en lingüística. |

## 1.3 Herramientas, métodos y metodologías

En esta sección, se detalla las herramientas y métodos a utilizar para obtener los resultados esperados indicados en la sección 1.2.3. En la

**Tabla 7**, se muestra de manera resumida las herramientas y métodos por resultado esperado; luego, se detalla cada uno de ellos. Finalmente, se explica la metodología a seguir para este proyecto de fin de carrera.

**Tabla 7** Herramientas y métodos a utilizar por resultado esperado

| Resultado Esperado | Herramientas, métodos y metodologías |
|---|---|
| **R1.** Software que realiza la extracción de las palabras en shiwilu (**O1**) | • Python<br>• MongoDB<br>• NLTK |
| **R2.** Listado de palabras shiwilu en su forma base y flexionada (**O1**) | • Python<br>• MongoDB<br>• NLTK |
| **R3.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Morfessor (**O2**)<br><br>**R4.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado a Linguistica (**O2**)<br><br>**R5.** Modelos automáticos no supervisados para la segmentación morfológica monolingüe con la selección de los parámetros más eficaces aplicado al método de Byte Pair Encoding (**O2**) | • Morfessor<br>• Linguistica<br>• Byte Pair Encoding<br>• MongoDB<br>• KDD<br>• NLTK<br>• *Bootstrapping with replacement* |

| Resultado Esperado | Herramientas, métodos y metodologías |
|---|---|
| **R6.** Segmentador morfológico automático a través de enfoque multilingüe haciendo uso de recursos en shiwilu e inglés (**O3**) | • Python<br>• NLTK<br>• Numpy<br>• FastText<br>• SequenceMatcher<br>• MongoDB<br>• SpaCy<br>• KDD<br>• *Bootstrapping with replacement* |
| **R7.** Selección del enfoque más preciso para segmentar morfológicamente la lengua shiwilu (**O4**) | • *Precision*<br>• *Recall*<br>• *F-measure*<br>• *Bootstrapping with replacement*<br>• Prueba Kolmogorov-Smirnov<br>• Prueba Barlett<br>• Prueba Welch ANOVA<br>• Prueba de múltiples comparaciones con el mejor (Hsu's MCB) |

### 1.3.1 Morfessor

Es una herramienta con método no supervisado para segmentar palabras en morfemas. La idea es descubrir una descripción de los datos tan compacta como sea posible. Las subcadenas que aparecen con frecuencia en diferentes formaciones de palabras se proponen como morfos y las palabras se representan como una concatenación de morfos (Bhat, 2012). Este modelo fue elegido porque permite el análisis de lenguas aglutinantes con morfología compleja a través de aprendizaje no supervisado (Bhat, 2012).

### 1.3.2 Linguistica

Es una herramienta que implementa el método de aprendizaje automático no supervisado de la morfología centrándose en la idea de la longitud mínima de descripción (MDL). MDL es una combinación del tamaño de la morfología (en términos teóricos de la información) y el tamaño de los datos comprimidos (o la longitud comprimida del corpus, dada por las probabilidades derivadas de la morfología). La heurística de aprendizaje avanza paso a paso para descubrir los sufijos candidatos básicos del lenguaje utilizando información mutua ponderada, usándolos para encontrar un conjunto de sufijos y luego usando MDL para corregir los errores generados por la heurística (Bhat, 2012). Este algoritmo fue elegido porque proporciona la lista de prefijos y sufijos con la información de frecuencia de aparición sin conocimiento previo del idioma (Bhat, 2012).

### 1.3.3 Byte Pair Encoding

BPE es una técnica de compresión de datos que fusiona pares frecuentes de bytes, en este caso, adaptada para la segmentación de palabras, fusionando caracteres o secuencias de caracteres. Este algoritmo es una estrategia de segmentación utilizada en la traducción automática, el cual ha demostrado buenos resultados en la traducción de palabras desconocidas (Sennrich, Haddow, & Birch, 2015). Fue elegido por permite extraer secuencias de caracteres de longitudes variables que dan como resultado la extracción de morfemas.

### 1.3.4 Métricas

Para comparar ambos enfoques, se utilizarán las métricas *Recall* y *Precision*, las cuales son usadas regularmente para medir el desempeño en la recuperación de información. Ambas se combinan en una sola métrica representando la media armónica conocida como *F-measure* (Makhoul, Kubala, Schwartz & Weischedel,1999).

Estas métricas se definen como:

- **Métrica de Precisión (*Precision)***

  Es el cociente entre el número de sufijos válidos encontrados y el total de sufijos encontrados (válidos e inválidos) (Shalonova et al., 2009). Esta métrica fue elegida porque puede estimarse de manera más confiable con recursos limitados (Baumann & Pierrehumbert, 2014).

- **Métrica de Sensibilidad o Exhaustividad (*Recall)***

  Es el cociente entre el total de sufijos identificados y el número de sufijos válidos existentes (Shalonova et al., 2009). Esta métrica fue elegida porque es utilizada en reconocimiento de patrones y permite identificar qué tan perfecta es la identificación de sufijos.

- **Valor F (*F-measure*)**

  Es el valor de la media armónica entre las métricas *Precision* y *Recall* (Shalonova et al., 2009). Esta métrica fue elegida para verificar cuál de las métricas *Precision* o *Recall* es más relevante.

Y se calculan de la siguiente manera:

$$Precision = \frac{C}{M}$$

$$Recall = \frac{C}{N}$$

$$F - measure = \frac{2\ (Precision\ \times Recall)}{(Precision + Recall)}$$

Donde:

- C: Número de valores correctos – valores identificados que comparados con los reales son calificados como correctos.
- M: Número total de valores identificados
- N: Número total de valores reales que se tiene como referencia o base

### 1.3.5 MongoDB

Es una base de datos NoSQL escalable y flexible que almacena los datos en documentos similares a JSON (Mongodb.com, 2014). Se ha elegido esta base de datos por la facilidad de acceso y manipulación de datos, así como su fácil integración en el desarrollo de código.

### 1.3.6 Python

Python es un lenguaje de programación de código abierto orientado a objetos. Se ha elegido este lenguaje porque es muy usado para exploración de datos e identificación de patrones en el aprendizaje automático; además, cuenta con varios paquetes para este fin (Swamynathan, 2017).

### 1.3.7 NLTK

Natural Language Tool Kit o NLTK es una librería para procesamiento de lenguaje natural. Es la más popular en Python y con mayor cantidad de contribuidores en el 2016. Esta librería es de código abierto bajo la licencia de Apache. Se eligió NLTK porque ha sido creada para apoyar la investigación y desarrollo en el Procesamiento de Lenguaje Natural (Swamynathan, 2017).

### 1.3.8 Spacy

Al igual que NLTK es una librería para procesamiento de lenguaje natural y se encuentra dentro de las más populares en Python. Está construida para una implementación eficiente de conceptos de procesamiento de lenguaje natural (Swamynathan, 2017). Se ha

elegido esta librería para utilizar los modelos pre-entrenados, y así identificar características del inglés útiles para el análisis multilingüe.

### 1.3.9 FastText

Es una librería de código abierto desarrollada por el Laboratorio de investigación de inteligencia artificial de Facebook. Esta librería permite realizar representaciones de texto a través de la información de subcadenas de palabras (Bojanowski, Grave, Joulin & Mikolov, 2016). Esta librería se ha elegido para realizar las representaciones vectoriales de las palabras en shiwilu e inglés.

### 1.3.10 SequenceMatcher

Basado en el algoritmo de reconocimiento de patrones de Ratcliff y Obershelp. SequenceMatcher es una clase de la librería difflib de Python que permite comparar pares de secuencias. Busca la subsecuencia de caracteres contigua más larga a través de la comparación del doble del número de caracteres coincidentes dividido entre el número total de caracteres de ambas cadenas. La similitud se mide a través de un ratio que varía entre 0 y 1, donde se considera que las cadenas tienen estrecha similitud si el ratio es mayor a 0.6. (Python Documentation, 2012). Esta librería se ha elegido para reconocer palabras o morfemas similares.

### 1.3.11 KDD (Knowledge Discovery in Databases)

KDD es el proceso de identificación de patrones válidos, novedosos, potencialmente útiles y comprensibles respecto a los datos. Este proceso consiste en una secuencia iterativa de pasos como se muestra en la **Figura 1** (Fayyad et al., 1996):

1. Identificación del dominio de aplicación y el conocimiento previo relevante; e identificar el objetivo del proceso KDD
2. Selección de datos objetivo

3. Limpieza y pre-procesamiento de datos
4. Transformación de datos
5. Minería de datos
6. Evaluación de patrones
7. Presentación del conocimiento

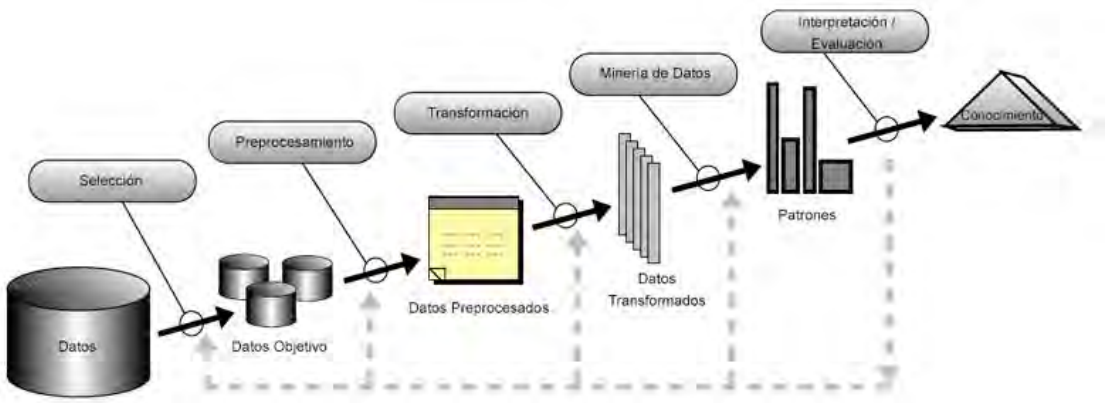


**Figura 1** Pasos del proceso KDD. Adaptado de Fayyad et al. (1996)

Se utilizan diferentes formas de procesamiento de datos donde son preparados para la minería. El paso de minería de datos puede interactuar con el usuario o una base de conocimiento. Los patrones interesantes se presentan al usuario y pueden almacenarse como nuevos conocimientos en la base del conocimiento (Han et al., 2011).

Este proceso fue elegido porque aborda el mapeo de datos de bajo nivel en otras formas más compactas, por ejemplo, un informe breve; más abstractas, por ejemplo, un modelo del proceso que generó los datos; o más útil, por ejemplo, un modelo predictivo para estimar el valor de casos futuros. Por lo tanto, permite que los descubrimientos sean más comprensibles para las personas (Fayyad et al., 1996).

### 1.3.12 Boostrapping with replacement

Es una técnica empleada para la estimación estadística sobre una población a través de varias muestras con reemplazo, es decir, un dato puede estar incluido en la muestra 0 o más

veces. Se utiliza en el aprendizaje automático para estimar las destrezas de modelos automáticos que realicen predicciones (Brownlee, 2018).

### 1.3.13 Prueba Kolmogorov-Smirnov

Es una prueba no paramétrica que permite probar que una muestra sigue una distribución normal. Se utiliza en muestras aleatorias de población continua. Plantea las siguientes hipótesis (Marques, 2001):

- $H_0$: Los datos provienen de una distribución normal.
- $H_1$: Alguno de los datos no corresponden a una distribución normal.

### 1.3.14 Prueba Barlett

Es una prueba estadística que permite probar la homogeneidad de varianzas entre dos o más muestras. Se utiliza sobre muestras con distribución normal. Plantea las siguientes hipótesis (Abell, Braselton & Rafter, 1999):

- $H_0$: Las muestras tienen la misma varianza.
- $H_1$: No todas las muestras tienen la misma varianza.

### 1.3.15 Prueba Welch ANOVA

Es un prueba aproximada de *one-way* ANOVA (*Analysis of variance*), que puede ser usada para comparar medias entre dos o más muestras con diferentes varianzas pero que sigan distribuciones normales. Esta prueba se realiza antes de identificar cuál es la muestra con la media más alta (Weerahandi, 2013).

### 1.3.16 Prueba Hsu's MCB

La prueba Hsu's MCB o de comparaciones múltiples con el mejor es una prueba *post hoc* que crea intervalos de confianza, donde compara las medias de las muestras con la

“mejor” media. La “mejor” media se define como la media más alta de las muestras. Es utilizada en muestras de tamaños iguales unto con las pruebas ANOVA (De Muth, 2014).

## 1.4 Alcance, limitaciones y riesgos

### 1.4.1 Alcance

Este proyecto de fin de carrera se enmarca en el área de Ciencias de la Computación, específicamente, en la rama de Procesamiento de Lenguaje Natural, abarcando únicamente el estudio computacional de la lengua como sistema a nivel morfológico, descartando otros niveles como fonológico, fonético, sintáctico, léxico o semántico. Específicamente, consiste en identificar un segmentador morfológico que permita identificar automáticamente los morfemas de palabras en la lengua shiwilu. Se tomó esta lengua como caso de estudio por ser de escasos recursos, pero de la cual se cuenta con algunos recursos para poder realizar este análisis.

La segmentación morfológica se realiza a través de dos enfoques. El enfoque monolingüe utiliza únicamente recursos de la lengua shiwilu; mientras que el enfoque multilingüe utiliza recursos de la lengua shiwilu e inglés, a partir de ejemplos traducidos que se encuentran en el diccionario que se usó como recurso base, brindado por la experta Pilar Valenzuela, sin generar nuevos recursos a partir de este.

En una interfaz web, se implementará el segmentador morfológico con el enfoque de mejor desempeño reflejado en las métricas seleccionadas de la comparación. Además, junto con él, se muestra el significado en español e inglés, así como algunos ejemplos en la lengua shiwilu con sus respectivas traducciones en español e inglés.

### 1.4.2 Limitaciones

Este proyecto está limitado por la falta de conocimiento de la lengua shiwilu; por ello, se cuenta con el apoyo de la profesora Pilar Valenzuela[1], quien es una de las principales investigadoras en lingüística sobre dicha lengua y la familia lingüística Pano, en general. Además, no se cuenta con un diccionario de shiwilu oficial, solo un avance del trabajo de campo realizado hasta agosto 2017.

### 1.4.3 Riesgos

En la

**Tabla 8**, se muestra una evaluación de los posibles riesgos para el proyecto de fin de carrera.

**Tabla 8** Riesgos del proyecto de fin de carrera

| Riesgo | Tipo de Riesgo | Impacto | Estrategia de mitigación |
|---|---|---|---|
| Muy pocos recursos digitalizados disponibles en la lengua shiwilu | Aceptable | La segmentación morfológica no podría realizarse con la misma eficiencia | Generar recursos en la lengua shiwilu a partir de datos previos recolectados |

[1] Profesora en Chapman University: https://www.chapman.edu/our-faculty/pilar-valenzuela

| Riesgo | Tipo de Riesgo | Impacto | Estrategia de mitigación |
|---|---|---|---|
| Falta de disponibilidad y de número de expertos en la lengua | Moderado | Los resultados de la segmentación morfológica no serían validados por un experto en la lengua | Fortalecer el vínculo que comprometa al experto en la lengua a realizar la validación de los resultados.<br><br>Utilizar otros recursos como diccionarios para verificar los resultados de la segmentación. |

## 1.5 Justificación

Desde una perspectiva computacional, los lingüistas interesados en el estudio de esta lengua se beneficiarán, a través de recursos digitales, es decir, gracias a datos digitalizados y herramientas automáticas de segmentación.

Además, este proyecto beneficiará a investigadores en lingüística computacional, pudiendo tomarlo como base para futuros estudios en la segmentación morfológica de otras lenguas con características similares (lengua de escasos recursos y lengua aglutinante). También, puede ser punto de partida para otros proyectos más complejos de procesamiento de lenguaje natural, como la implementación de traductores automáticos.

Finalmente, este proyecto apoyará en la revitalización y difusión de la lengua shiwilu, permitiendo que siga viva y fomentando el interés de sus hablantes.

# Capítulo 2. Marco Conceptual

En esta sección se definirán conceptos relacionados a la lingüística y computación, que faciliten la comprensión y entendimiento del problema de complejidad en la segmentación morfológica de una lengua de escasa documentación para este proyecto de fin de carrera.

## 2.1 Morfología

La morfología es la parte de la gramática que estudia la estructura interna de las palabras, las variantes que estas presentan, los segmentos que la componen y la forma en que estos se combinan (Real Academia Española, 2016).

## 2.2 Morfema

El morfema o segmento morfológico se define como la unidad mínima aislable en la segmentación morfológica. Por ejemplo, en la palabra *habilidades* se identifican los siguientes morfemas: la raíz ***habil***–, que aporta el significado léxico, presente en *habilitar, rehabilitar, habilidoso*, etc.; el sufijo derivativo –***idad***, identificable en *claridad, cordialidad, felicidad, etc.*; y el sufijo flexivo –***es***, que se encuentra en *carteles, mujeres, felices*, etc. (RAE, 2016).

Los morfemas reciben distintos nombres en función del papel que desempeñan en la estructura y en el proceso de formación de las palabras (RAE, 2016). Las **raíces** forman el núcleo de la palabra, a las cuales se unen otras partes. Los **afijos** son pequeños morfemas con significado abstracto, pero sin valor por sí solos (Haspelmath & Sims, 2010). Estos aparecen unidos a la raíz o a otro morfema y se clasifican según su posición respecto a la raíz (RAE, 2016):

- Sufijos, si se une al final de la raíz. Por ejemplo: *habil-idad*
- Prefijos, si se une al principio de la raíz. Por ejemplo: *im-posible*
- Interfijos, situados entre la raíz y un sufijo. Por ejemplo: *polv-ar-eda*

## 2.3 Alomorfo

Según Haspelmath (2010), se utiliza el término alomorfo para hacer referencia a un mismo afijo que puede tener múltiples formas o realizaciones. Es frecuente que un afijo tenga múltiples alomorfos, haciendo que el análisis sea más complejo. Una de las complicaciones más frecuentes es que los morfemas pueden tener diferentes formas fonológicas bajo diferentes circunstancias. Por ejemplo: en inglés, el morfema que denota el plural puede pronunciarse de 3 maneras:

- [s] (como en cats [kæts])
- [z] (como en dogs [dɒgz])
- [-əz] (como en faces [feisəz])

### 2.3.1 Segmentación Morfológica

Segmentar significa separar palabras complejas en partes que contengan un significado. En este caso, estas partes son los morfemas. Las palabras pueden ser segmentadas en más de dos morfemas. Sin embargo, en el caso que una palabra no pueda separarse en morfemas, nos referimos a un **monomórfico** (Haspelmath & Sims, 2010).

### 2.3.2 Lengua de escasos recursos

Para este término también son usados los siguientes nombres: lenguas menores, lenguas minoritarias, lenguas menos usadas, pequeñas lenguas, lenguas más pequeñas, lenguas pobres en recursos, lenguas de menos recursos, etc. (Forcada, 2006).

Esta lengua es usada en un país por un pequeño grupo de la población. Estos hablantes se diferencian por sus características étnicas, religiosas o culturales, las cuales desean preservar (Crystal, 2011). La conservación de la lengua es el principal objetivo de estas comunidades. En general, los miembros de estas comunidades no dominan la lengua oficial y no adquieren la cultura general del país, lo que conlleva a dificultades en su integración

política y económica respecto a la comunidad mayoritaria, sin poder tomar decisiones en ella (Fase, Jaspaert & Kroon, 1992).

Desde el punto de vista computacional, estas lenguas tienen presencia limitada en Internet y no cuentan con recursos legibles para las máquinas que dependan de tecnologías externas como datos lingüísticos, corpus, etc. (Forcada, 2006).

### 2.3.3 Lengua aglutinante

Es una lengua en la cual varios afijos pueden ser añadidos a la raíz de una palabra, modificando su significado o función gramatical. Entre las lenguas aglutinantes más conocidas tenemos: finlandés, húngaro, swahili y turco. (Richards & Schmidt, 2013).

En América del Sur, las lenguas son consideradas aglutinantes, tal es el caso del quechua y el aymara; sin embargo, la aglutinación varía para cada lengua en particular (Campbell & Grondona, 2012). En la **Tabla 9**, se muestra un ejemplo sobre la morfología en una lengua aglutinante, en este caso quechua (Cerrón-Palomino, 1994).

**Tabla 9** Ejemplo de morfología compleja en una lengua aglutinante. Adaptado de Cerrón-Palomino (1994)

| *tarpu-ysi-ri-chi-ku-naya-wa-sqa-yki-chik-manta-lla-ña-puni-chá*<br>"Seguramente, pues, desde que Uds. trataron de que yo sienta deseos de ayudarles nomás a sembrar" | | | |
|---|---|---|---|
| *tarpu-* | *-ysi-ri-chi-ku-naya-wa-sqa* | *-yki-chik-manta* | *-lla-ña-puni-chá* |
| RAIZ | Sufijos derivativos | Sufijos flexivos | Sufijos independientes |

En este caso, se tiene una palabra con estructura sumamente elaborada, donde cada morfema es separado por un guión; se puede afirmar que la morfología es el componente más complejo de la gramática de este tipo de lenguas (Cerrón-Palomino, 1994).

### 2.3.4 Lengua fusionante

Es una lengua donde los afijos no son fáciles de identificar como en las lenguas aglutinantes, debido a que estos se fusionan entre ellos y la raiz, haciendo que un solo morfema pueda almacenar más de un significado gramatical, sintáctico y semántico. Algunas lenguas fusionantes son: griego, latín, español, ruso, francés e italiano (Wheeldon, 2002).

### 2.3.5 Aprendizaje automático

El aprendizaje automático es "la programación de computadoras para optimizar un criterio de rendimiento utilizando datos de ejemplo o la experiencia pasada" (Alpaydin, 2010). Así, el objetivo principal es que esta experiencia permita construir un modelo computacional a través del estudio de la adquisición automática de conocimiento para la mejora del rendimiento de los sistemas. Esto permite la mejora continua de sí mismo y con ello, mayor eficiencia y efectividad (Management Association, 2011).

También, se trata de que las computadoras modifiquen o adapten sus acciones, de tal manera que estas acciones sean más precisas. La certitud con la que se eligen estas acciones refleja esta precisión (Marsland, 2015).

Por una parte, para poder realizar inferencias a partir de una muestra, se utiliza la teoría estadística. Por otra parte, la ciencia de la computación cumple un doble rol: a través de algoritmos eficientes para resolver el problema de manera óptima y de la eficiencia en el modelo aprendido (Alpaydin, 2010).

En el proyecto, se utiliza el aprendizaje automático para poder inducir rápidamente los morfemas de la lengua shiwilu, y permitiendo que el modelo aprendido pueda seguir identificándolos.

### 2.3.6 Aprendizaje supervisado y no supervisado

El tipo de aprendizaje automático más común es el aprendizaje supervisado. También conocido como aprendizaje por ejemplares, ya que se basa en un conjunto de entrenamiento con muestras y respuestas correctas, el cual generaliza un algoritmo para responder de manera correcta (Marsland, 2015).

Por otra parte, el aprendizaje no supervisado intenta identificar las similitudes entre los datos de entrada para poder ser clasificados juntos, puesto que no tiene inicialmente el conjunto de entrenamiento (Marsland, 2015).

Es decir, el aprendizaje supervisado se realiza a través de un mapeo de datos de entrada y cuya salida es validada por un supervisor. Mientras que, en el aprendizaje no supervisado, solo se cuenta con datos de entrada y sin supervisor, basándose en patrones que ocurren con mayor regularidad que otros (Alpaydin, 2010).

En el proyecto, la segmentación bajo los enfoques monolingüe y multilingüe es no supervisado debido a los escasos recursos y a la falta de información relevante relacionada con la morfología de la lengua shiwilu, teniendo así solo datos de entrada.

### 2.3.7 Procesamiento de lenguaje multilingüe

Para el caso en el que las lenguas tienen escasos recursos, es posible utilizar el enfoque multilingüe, el cual involucra aprender información de múltiples lenguas de manera simultánea. Este enfoque ha obtenido mejores logros al aumentar el número de lenguas en estudio (O'Horan et al., 2016).

Las soluciones más populares de procesamiento de lenguaje multilingüe son transferencia de información de lenguas de abundantes recursos a lenguas de escasos recursos, aprendizaje multilingüe conjunto, y desarrollo de modelos universales (O'Horan et al., 2016).

# Capítulo 3. Estado del Arte

Esta sección presenta investigaciones realizadas para el problema de la complejidad en la segmentación morfológica de las palabras de una lengua de escasos recursos.

## 3.1 Revisión sistemática

El estudio del estado del arte se realizó a través del método de revisión sistemática utilizando PICOC.

### 3.1.1 Formulación de la pregunta

Este estudio abordará las siguientes preguntas de investigación:

- ¿Qué implementaciones se realizaron a cabo respecto a la segmentación morfológica de una lengua de pocos recursos a través del aprendizaje automático no supervisado?
- ¿Bajo qué condiciones se realizaron estas implementaciones (tipo de lengua, cantidad de recursos utilizados, métodos utilizados)?

### 3.1.2 Palabras clave

En base a la pregunta formulada, se determinaron las siguientes palabras clave y algunos términos alternativos:

- *Unsupervised learning*
- *Morphology*
- *Agglutinative*
- *Under-resourced* / *low-resourced* / *minoritary* / *endangered*
- *Multilingual analysis* / *Cross-lingual analysis*
- *Morphological analyzer*

### 3.1.3 Cadenas de búsqueda

A través de las palabras clave se obtuvo la siguiente cadena de búsqueda:

*"unsupervised learning" AND "morphology" AND "analyzer" AND "agglutinative" AND ("under-resourced" OR "low-resourced" OR "minoritary" OR "endangered" OR "multilingual")*

### 3.1.4 Estrategia de búsqueda

Dada la problemática, el objetivo principal fue realizar una búsqueda de analizadores o segmentadores morfológicos con énfasis en lenguas de escasos recursos; utilizando aprendizaje automático no supervisado. Por otra parte, como en el caso de estudio se tendrá en cuenta la lengua shiwilu, también se tiene preferencia por los estudios aplicados a lenguas aglutinantes. El proceso de búsqueda se realizó a través de Google Scholar, debido a su libre acceso y amplia cobertura de literatura, haciendo uso de la cadena de búsqueda planteada.

### 3.1.5 Selección de artículos y criterios de exclusión

Se seleccionaron artículos publicados en conferencias, revistas indexadas o repositorios con revisores entre pares, los cuales contenían las palabras clave en el título y/o resumen. Se excluyeron artículos no relacionados a la rama computacional. Así, se encontraron 131 artículos, de los cuales 80 no repetidos, seleccionando 6 artículos con contenido más detallado y relevante para el estudio donde se realize segmentación morfológica en lenguas aglutinantes de escasos recursos a través de enfoques monolingües o multilingües no supervisados.

## 3.2 Resultados de la revisión

### 3.2.1 Aprendizaje morfológico para lenguas fusionantes y aglutinantes de pocos recursos

Según Shalonova, Golénia y Flach (2009), se realiza una segmentación morfológica tomando como casos la lengua rusa (lengua fusionante) y la lengua turca (lengua aglutinante). Se presentan 2 algoritmos para la segmentación morfológica. El primero de ellos llamado el algoritmo *Tree of Aligned Suffix Rules* o TASR. Este algoritmo compara el par de palabras, generando un conjunto de posibles reglas de sufijos. También puede ser utilizado para extraer la raíz o tema de una palabra, con supervisión mínima.

El segundo algoritmo no supervisado, llamado *Graph-based Unsupervised Suffix Segmentation* o GBUSS, representa los sufijos encontrados en el corpus como nodos en un grafo de sufijos; mientras que las aristas dirigidas indican la sucesión de un sufijo por otro. Esto permite segmentar una cadena de sufijos en sufijos con significado gramatical.

Dado que se desea trabajar en lenguas de escasos recursos, se utilizaron pocos datos de entrenamiento. Para el primer algoritmo, se utilizó 1392 pares de palabras de verbos en inglés con su forma en tiempo pasado. También se utilizó las 1000 raíces más frecuentes para sustantivos rusos. Como tercer recurso, se utilizó 4000 pares de palabras de participio en ruso. Mientras que el segundo algoritmo utilizó 200 sufijos únicos extraídos de participios rusos, además de cuatro criterios de parada para prevenir una generalización en exceso, que podría hacer que pequeños sufijos no sean identificados, al estar contenidos en otros más grandes. También se realizó una prueba con el algoritmo GBUSS; utilizando 3000 sustantivos y 2000 verbos turcos, los cuales son ejemplos de morfología derivacional.

Los resultados fueron validados frente a los resultados obtenidos por un sistema llamado Morfessor, el cual utiliza dos métodos no supervisados. El primero, basado en un

método recursivo para el enfoque *minimal description length* (MDL); y el segundo, un método secuencial para máxima probabilidad usando búsqueda Viterbi que permite una segmentación óptima (Creutz & Lagus, 2005).

El trabajo realizado permite demostrar que es factible procesar lenguas de escasos recursos fusionantes o aglutinantes. Para el caso de lenguas aglutinantes, se puede obtener resultados con métodos no supervisados.

### 3.2.2 Aprendizaje de morfología semi-supervisada para lenguas aglutinantes usando conjuntos de entrenamiento pequeños

Según Shalonova y Golénia (2010), este trabajo realiza la segmentación morfológica de lenguas aglutinantes, en este caso, turco y zulú, con una pequeña cantidad de supervisión. Se utilizó de una lista de palabras más frecuentes: 1457 verbos y 2267 sustantivos en lengua turca; además, 846 sustantivos y 931 verbos en zulú.

Primero se utiliza un algoritmo supervisado, llamado *Supervised Stem Extraction* o SSE, que permite la extracción de la raíz de una palabra; a partir de una etapa de entrenamiento con un conjunto de palabras cuyos límites de la raíz están señalados. Luego, se utiliza el algoritmo no supervisado GBUMS (*GraphBased Unsupervised Morpheme Segmentation*), para segmentar sufijos y prefijos.

Los resultados de los experimentos fueron comparados con el sistema Morfessor, considerado uno de los mejores en aprendizaje automático de morfología no supervisada. Se obtuvo gran rendimiento considerando que se desea trabajar con lenguas de recursos limitados.

### 3.2.3 Lenguas de abundantes recursos para la mejora de análisis morfológico de lenguas de escasos recursos

Según Baumann y Pierrehumbert (2014), este caso de estudio utiliza con gran precisión los recursos disponibles de una lengua de abundantes recursos, inglés, para el análisis morfológico de dos lenguas de escasos recursos, tagalo y zulú, que tengan algunos préstamos lingüísticos.

Para este experimento se utilizó una lista de palabras recopiladas de 80 horas de diálogos telefónicos. De esta lista, se utilizaron 16 655 palabras de tagalo y 54 009 palabras en zulú. Además, se utiliza una lista de palabras en inglés; por lo tanto, no es un método estrictamente no supervisado, a pesar de no necesitar datos etiquetados.

Los resultados fueron comprobados con una referencia gramatical de tagalo y un pequeño corpus de zulú. Consiguiendo efectivamente 20 prefijos flexivos, 1 prefijos derivacional y 3 infijos para tagalo; así como, 9 sufijos y 57 prefijos para zulú.

### 3.2.4 Segmentación morfológica de la lengua Kannada

Según Bhat (2012), Kannada es una lengua de escasos recursos con morfología flexiva y aglutinante. Se utilizaron dos corpus: una colección de cuentos infantiles y un conjunto de ensayos de diversos temas; de las cuales se extrajeron 24 851 y 210 368 palabras, respectivamente.

Se utilizaron 3 métodos. El primero, Linguistica, una herramienta de aprendizaje no supervisado desarrollado por Goldsmith; el segundo, Morfessor, un segmentador morfológico no supervisado; y un tercer algoritmo, UnDivide, un segmentador morfológico independiente del lenguaje basado en el algoritmo no supervisado de Keshava y Pitler.

Todos los métodos se ejecutaron por separado sobre los dos corpus. Los resultados muestran que UnDivide muestra un mayor desempeño para sustantivos flexivos en el

conjunto de ensayos; mientras que Morfessor, para las palabras de cuentos infantiles. Además, UnDivide muestra mayor desempeño para ambos corpus considerando solo verbos flexivos.

### 3.2.5 Analizador morfológico de lengua uigur

La lengua uigur es una lengua aglutinante cuya mayor dificultad son sus alteraciones fonéticas. Se seleccionaron 18 400 palabras y se utilizaron dos enfoques de segmentación con método supervisado basado en reglas y un modelo estadístico. Los resultados de segmentación de 9 025 oraciones se verificaron de manera manual con un 97.66% de precisión (Ablimit, Kawahara, Pattar & Hamdulla, 2016).

### 3.2.6 Aprendizaje no supervisado de morfología aglutinante utilizando el proceso Pitman-Yor

Este estudio se realizó en dos lenguas aglutinantes: Malayalam y Kannada. En ambas lenguas se observa que se pueden encontrar dos o más palabras de manera conjunta; entonces, en una cadena de texto se podría encontrar el sujeto y el verbo a la vez. Se utiliza el proceso Pitman-Yor anidado para segmentar en palabras. Luego, se realiza la segmentación morfológica a través de un algoritmo de identificación y evaluación de manera no supervisada.

Ya que ambas lenguas son de escasos recursos, utilizan corpus de Wikipedia con 10 000 palabras y una lista de sufijos y prefijos de Wiktionary para refinar los morfemas identificados. Los resultados fueron comparados con el sistema Morfessor. Sin embargo, ambos métodos fracasaron en la identificación correcta de los límites del morfema; ya que, en estas lenguas, son muy frecuentes los cambios morfofonémicos (Kumar, Padró & Oliver, 2015).

## 3.3 Conclusiones

De acuerdo a la revisión del estado del arte, se observa el interés por el aprendizaje no supervisado de la morfología de una lengua. Se plantean diversas posibles soluciones dada la naturaleza de la lengua (aglutinante) y sus escasos recursos. Los estudios realizados se resumen en la **Tabla 10**.

**Tabla 10** Cuadro resumen de la revisión del estado del arte

| Estudio | Dominio (lenguas) | Datos utilizados | Métodos utilizados | Componentes resultados |
|---|---|---|---|---|
| 1. Aprendizaje morfológico para lenguas fusionantes y aglutinantes de pocos recursos (Shalonova, Golénia y Flach, 2009) | Ruso y turco | - 1392 pares de palabras de verbos en inglés<br>- 1000 raíces y 4000 pares de palabras de participio en ruso.<br>- 200 sufijos de participios rusos<br>- 3000 sustantivos y 2000 verbos turcos | 2 algoritmos no supervisados:<br>- Tree of Aligned Suffix Rules o TASR<br>- Graph-based Unsupervised Suffix Segmentation o GBUSS | Segmentador morfológico para lengua fusionante.<br>Segmentador morfológico para lengua aglutinante |
| 2. Aprendizaje de morfología semi-supervisada para lenguas aglutinantes usando conjuntos de entrenamiento pequeños (Shalonova y Golénia, 2010) | Turco y zulú | - 1457 verbos y 2267 sustantivos turcos<br>- 846 sustantivos y 931 verbos zulú | Algoritmo supervisado Supervised Stem Extraction o SSE.<br>Algoritmo no supervisado GraphBased Unsupervised Morpheme Segmentation o GBUMS | Segmentador morfológico para lenguas aglutinantes |

| Estudio | Dominio (lenguas) | Datos utilizados | Métodos utilizados | Componentes resultados |
|---|---|---|---|---|
| 3. Lenguas de abundantes recursos para la mejora de análisis morfológico de lenguas de escasos recursos (Baumann y Pierrehumbert, 2014) | Tagalo y Zulú | - 16 655 palabras en tagalo<br>- 54 009 palabras en zulú<br>- lista de palabras en inglés | Método (no estrictamente) no supervisado | Segmentador morfológico a partir de otra lengua con abundantes recursos. |
| 4. Segmentación morfológica de la lengua Kannada (Bhat, 2012) | Kannada | - 24 851 palabras en Kannada de una colección de cuentos infantiles<br>- 210 368 palabras en Kannada de un conjunto de ensayos de diversos temas | Algoritmos no supervisados:<br>- Morfessor<br>- UnDivide<br>- Linguistica | Resultados de un estudio comparativo de algoritmos no supervisados para segmentación morfológica |
| 5. Analizador morfológico de lengua uigur (Ablimit, Kawahara, Pattar & Hamdulla, 2016) | Uigur | - 18 400 palabras en uigur | Método supervisado basado en reglas y un modelo estadístico | Segmentador morfológico supervisado |
| 6. Aprendizaje no supervisado de morfología aglutinante utilizando el proceso Pitman-Yor<br>(Kumar, Padró & Oliver, 2015) | Malayalam y Kannada | - 10 000 palabras tomadas de Wikipedia<br>- Una lista de sufijos y prefijos de Wiktionary para refinar los morfemas identificados | Proceso Pitman-Yor anidado.<br>Algoritmo de identificación y evaluación no supervisado. | Segmentador de palabras y segmentador morfológico |

# Capítulo 4. Preprocesamiento de los recursos en shiwilu

En esta sección, se explica los pasos seguidos para la obtención de los datos iniciales y su pre-procesamiento. Este capítulo relacionado al objetivo específico 1, indica cómo se realizó la extracción de datos y las palabras shiwilu en su forma base y flexionada, siguiendo la metodología KDD.

Este paso es importante ya que se cuenta con un solo origen de los recursos de la lengua, debido a los pocos recursos digitalizados. Por lo tanto, es necesario este desarrollo para poder realizar la segmentación morfológica.

## 4.1 Extracción de datos

Previo al análisis morfológico, se recopiló 6884 entradas léxicas del diccionario shiwilu, con información relevante como definición, categoría gramatical y ejemplos, tanto en inglés como en español. En la **Figura 2**, se muestran los pasos seguidos, los cuales son detallados a continuación.


**Figura 2** Pasos previos para la obtención de datos relevantes para el proyecto

### 4.1.1 Origen de los datos

Como producto del trabajo de campo realizado por la profesora Pilar Valenzuela hasta agosto de 2017, se obtuvo un archivo XML con información recopilada a través de FieldWorks (https://software.sil.org/fieldworks/), el cual permite construir léxicos y textos interlineales para la publicación de diccionarios, como se observa en la **Figura 3**.

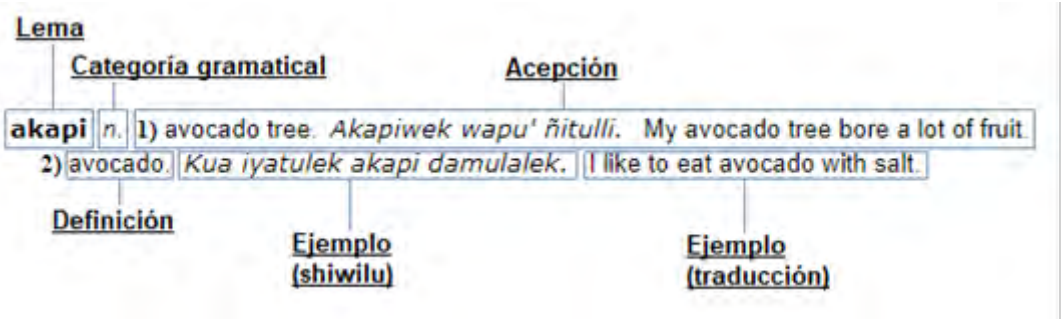


**Figura 3** Datos relevantes en diccionario shiwilu generado por FieldWorks

### 4.1.2 Selección de datos

Para poder identificar los datos a utilizar, se extrajo un árbol con la estructura de todas las posibles etiquetas a utilizar del archivo. A partir de este árbol, resumido en la **Figura 4**, se identificaron las etiquetas que contenían información relevante para el análisis. Por cada entrada léxica se consideraron las siguientes etiquetas.

- **LexEntry_HeadWord**

Indica el lema de la entrada léxica. En algunos casos puede tratarse de un afijo, estando marcados por el carácter guión "-" al inicio de la cadena para los sufijos y al final de la cadena para los prefijos. Por ejemplo, se encontraron entradas léxicas como "-a'ka'su" y "pada'-", los cuales son un sufijo y un prefijo, respectivamente.

- **LexEntry_Senses**

Cada una de sus subetiquetas **LexSense** contienen la información de alguno de los significados del lema. Un lema puede tener múltiples significados o acepciones.

  - **MoMorphSynAnalysisLink_MLPartOfSpeech.** Indica la categoría gramatical de la acepción. Pueden existir dos o más acepciones con la misma categoría gramatical.
  - **LexSense_Definition.** Declara y/o explica la acepción del lema según el contexto
  - **LexSense_Examples.** Dentro de esta etiqueta se encuentran ejemplos de la acepción. Una acepción puede contener múltiples ejemplos.

- **LexExampleSentence_Examples.** Ejemplifica una oración o frase en lengua shiwilu.
- **LexExampleSentence_Translations.** Muestra las traducciones de la oración o frase ejemplificadas en la etiqueta anterior. Se pueden encontrar traducciones en español y/o inglés.

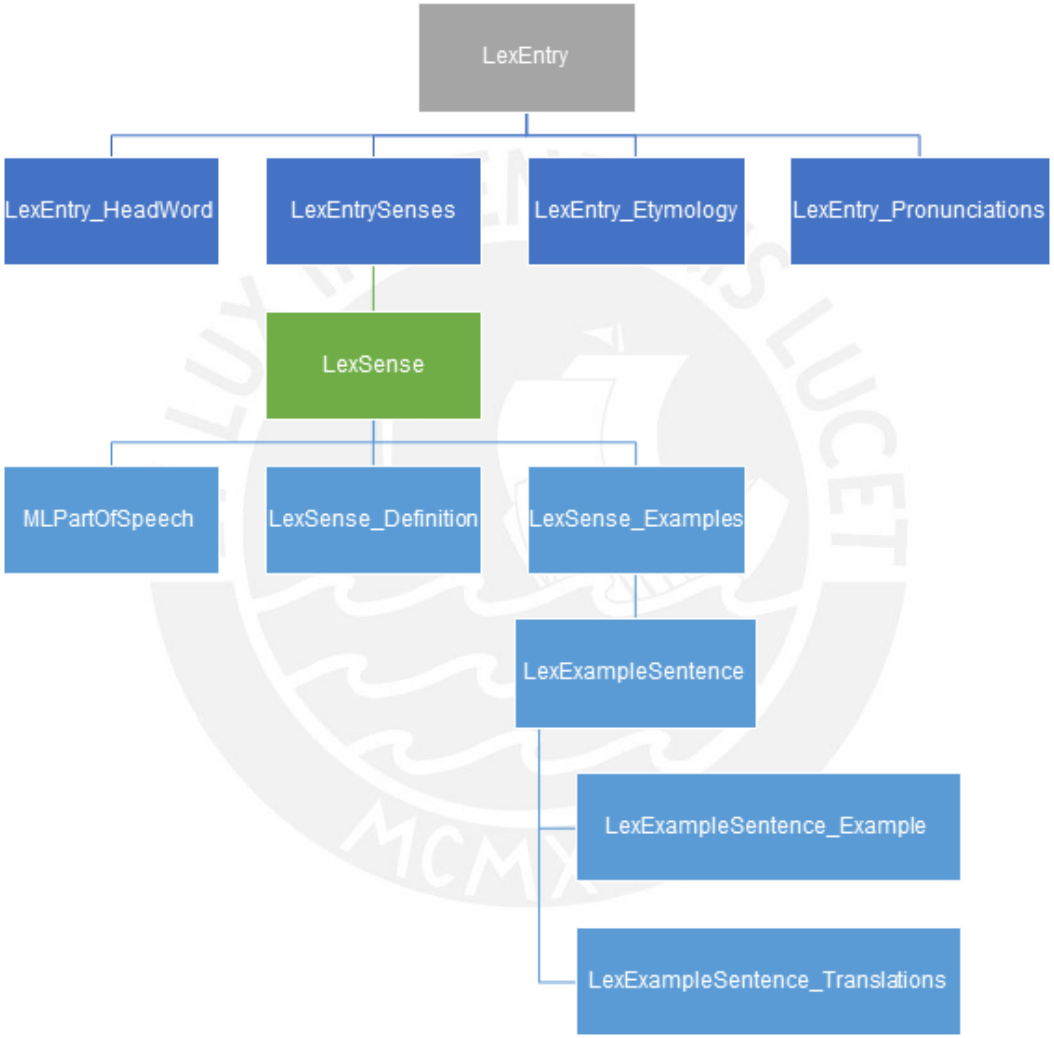


**Figura 4** Resumen de estructura XML de archivo de datos iniciales

Cabe resaltar que no todas las entradas léxicas tienen información en todas las etiquetas. Por ejemplo, existen palabras que solo indican el lema sin mayor información.

### 4.1.3 Limpieza de datos

Una vez identificados los datos a utilizar, se procedió con la limpieza de los datos, en algunos casos se encontraron marcas en los lemas y/o ejemplos como los caracteres '?' o '*'. También se separaron los afijos del resto de lemas para poder ser usados posteriormente en la validación de morfemas identificados.

### 4.1.4 Transformación de datos

La estructura de los datos se estandarizó, de tal manera que puedan ser fácilmente consultados, separando los lemas de sufijos y prefijos; evitando duplicados. También es posible identificar si está disponible la categoría gramatical, si tienen 1 o más acepciones, y a su vez si estos tienen algún ejemplo con o sin sus respectivas traducciones.

## 4.2 Almacenamiento de datos

Se crearon archivos JSON compuestos por listas de diccionarios estructurados, cuya forma se ejemplifica en la **Figura 5**, a partir de los datos estandarizados que posteriormente fueron almacenados en una base de datos MongoDB.


```
{
  "lemma": "a'dandektapalli",
  "category": "vt.",
  "senses": [
    {
      "es_def": "pescar con canasto.",
      "en_def": "to fish with a basket.",
      "examples": [
        {
          "shi": "Wilawek a'dandektulli wapu' pekta.",
          "es": "Mi hijo ha agarrado con canasto bastante mojarra.",
          "en": "My son caught a lot of mojarra fish with a basket."
        }
      ]
    }
  ]
}
```


**Figura 5** Estructura de diccionarios de archivo json creados para el almacenamiento en MongoDB

## 4.3 Resumen de datos extraídos

En la **Tabla 11**, se detalla en números los datos disponibles para su uso en el presente proyecto.

**Tabla 11** Resumen en números de datos extraídos

| Representación de datos | Cantidad |
|---|---|
| Total entradas léxicas | 6884 |
| Total definiciones | 5257 |
| Entradas que tienen, por lo menos, una definición (en inglés y/o español) | 4891 |
| Definiciones en español | 5254 |
| Definiciones en inglés | 5236 |
| Entradas que tienen, por lo menos, un ejemplo (en shiwilu, español y/o inglés) | 5393 |
| Ejemplos en shiwilu | 6718 |
| Ejemplos en español | 6714 |
| Ejemplos en inglés | 6716 |
| Sufijos encontrados (únicos) | 283 |
| Prefijos encontrados (únicos) | 30 |
| Entradas donde se indica la categoría gramatical | 4588 |

## 4.4 Recursos utilizados

Se procesaron algunos datos de acuerdo a lo necesario para cada tipo de segmentación y/o resultado esperado. En la **Figura 6** y **Figura 7**, se puede apreciar los recursos utilizados para la obtención de los diferentes resultados esperados.


**Figura 6** Recursos utilizados para el resultado esperado 3, 4 y 5

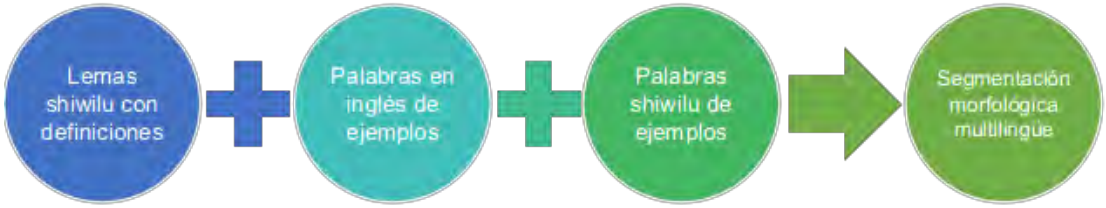


**Figura 7** Recursos utilizados para el resultado esperado 6

### 4.4.1 Pre-procesamiento para la segmentación monolingüe

Como se observa en la **Figura 6**, para el análisis morfológico monolingüe, se utilizó únicamente datos en lengua shiwilu. Para esto se tomaron en cuenta los lemas de las entradas léxicas, a excepción de las entradas cuya categoría gramatical es "nprop." (nombre propio), ya que es irrelevante para el análisis, por no contener información de la lengua en estudio.

Además, se utilizaron las oraciones de los ejemplos en shiwilu. Cada ejemplo en shiwilu fue segmentado en palabras. Se excluyeron todo tipo de signo, ya sea de puntuación, admiración, interrogación, etc. Sin embargo, se preservó el carácter de comilla simple que forma parte de las palabras en shiwilu.

De esta manera, se extrajo un conjunto de 13 377 palabras únicas en shiwilu en texto plano, siendo este el único recurso utilizado para la identificación de morfemas a través del enfoque monolingüe.

### 4.4.2 Pre-procesamiento para la segmentación multilingüe

El análisis morfológico multilingüe utilizó oraciones paralelas en shiwilu e inglés. Se considera relevante el uso de datos en inglés porque es una de las lenguas con mayores recursos disponibles. Así, se extrajeron 6716 oraciones en shiwilu con sus respectivas traducciones al inglés. Estas oraciones son usadas para generar los espacios vectoriales de cada lengua.

Adicionalmente, para alinear los espacios vectoriales, se generó automáticamente un diccionario bilingüe con la palabra en shiwilu y su respectiva traducción a partir de su definición. Para generar el diccionario, se tomó en cuenta 2 factores:

1. Se buscaron todas las definiciones en inglés que contengan 1 sola palabra, así se puede afirmar que estamos frente a la traducción exacta de una forma base de palabra (lema)
2. Se identificaron todas las palabras idénticas que están tanto en la oración en shiwilu como en su respectiva oración traducida, ya que estaríamos frente a palabras que no tienen una traducción exacta al inglés, tal es el caso de nombres propios y/o préstamos entre lenguas. Así, se formó un pequeño diccionario de 1024 palabras.

Para poder identificar las características morfológicas de las palabras en shiwilu, previamente se identificaron las características en inglés. Para ello, se utilizó la librería SpaCy, el cual tiene modelos pre-entrenados, pudiendo identificar sus características gracias a los recursos existentes en inglés.

# Capítulo 5. Análisis morfológico monolingüe

En este capítulo, correspondiente al objetivo específico 2 y detalla cómo se obtuvieron los parámetros más eficaces sobre los modelos monolingües. Tomando en cuenta los resultados del estudio realizado por Ács y Velkey (s.f.), se seleccionaron Morfessor, Linguistica y BPE para este análisis, ya que obtuvieron mejores resultados en la identificación automática de morfemas para conjuntos de datos pequeños. Por ello, se realizó el experimento en estos tres algoritmos sobre la lista de palabras en shiwilu descrita en 4.4.1.

## 5.1 Optimización de parámetros para Morfessor

En Morfessor, los datos de entrada para el algoritmo son llamados *compounds*, en este caso son las palabras en shiwilu, y devuelve *constructions*, es decir los morfemas que se desean identificar, los cuales están formados por uno o más *atoms*, pequeños fragmentos de texto que el algoritmo procesa, para este caso, letras.

*Morfessor* utiliza dos tipos de entrenamiento: *batch* y *on-line*. Para este caso, se utilizó el entrenamiento *batch* porque se cuenta con un solo conjunto de datos inicial cargados en su totalidad antes del entrenamiento como una lista de palabras. Este entrenamiento itera sobre los datos proporcionados al inicio, y a partir de este, se inducen los morfemas.

Los parámetros configurables para este entrenamiento son *algorithm* y *finish_threshold*. *Algorithm* indica el algoritmo que segmenta las palabras y *finish_threshold* indica hasta cuándo realizar las segmentaciones.

Para seleccionar los párametros a utilizar en este experimento, se realizaron 20 entrenamientos sobre el conjunto de 13 377 palabras shiwilu extraidas. Los conjuntos de parámetros utilizados se formaron a partir de las combinaciones de los valores indicados en la **Tabla 12**.

**Tabla 12** Posibles valores de parámetros de Morfessor

| Parámetro | Valor por defecto | Valores analizados | |
|---|---|---|---|
| algorithm | recursive | recursive<br>viterbi | |
| finish_threshold | 0.005 | 0.0000005<br>0.000005<br>0.00005<br>0.0005<br>0.005 | 0.05<br>0.5<br>5<br>50<br>500 |

Estos primeros experimentos se ejecutaron para evaluar el grado de influencia de los parámetros en los resultados. En la **Figura 8**, se muestra un gráfico de la métrica *F-measure* obtenidos en los 20 entrenamientos, cuyos resultados detallados se encuentran en el Error! Reference source not found.Error! Reference source not found.. En virtud a estos resultados, se podría inferir que, pese a la variación del parámetro *finish_threshold,* el algoritmo de segmentación *recursive* tiene mejores resultados que *viterbi*, ya que las métricas son más altas, por lo tanto, muestran mayor exactitud al identificar morfemas.

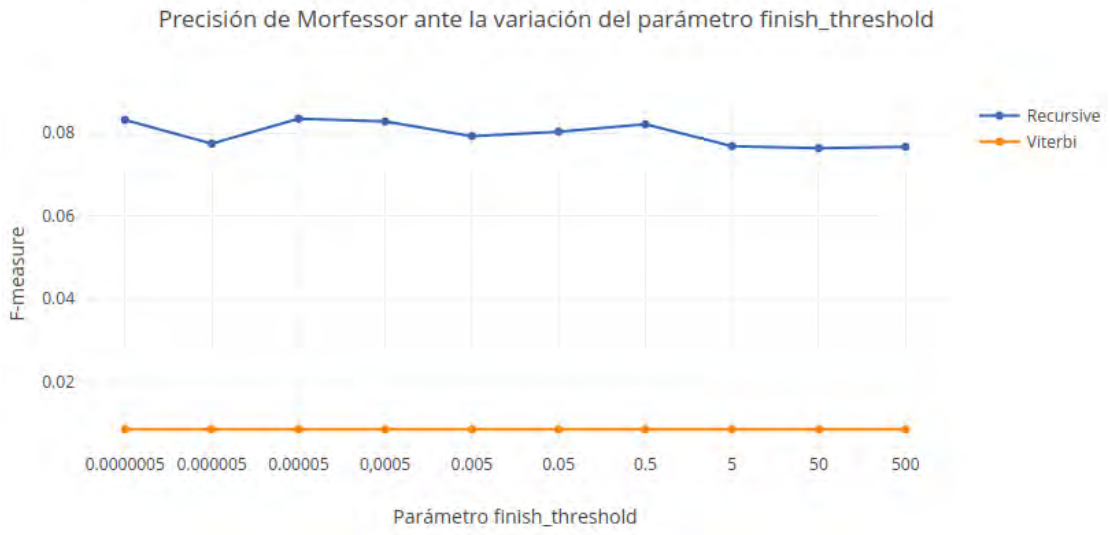


**Figura 8** Gráfico de resultados de *F*-measure para diferentes valores de parámetro *finish_threshold* en Morfessor

Para descartar el algoritmo de división *viterbi*, se utilizó el método de muestreo con reemplazo. Primero, se seleccionó la cantidad de submuestras y su tamaño. Se construyeron 30 submuestras aleatorias de 13 377 elementos que podrían estar repetidos a partir de la muestra original de palabras shiwilu. Es necesario resaltar que, de acuerdo a la literatura, la cantidad de submuestras elegidas es suficiente para este tipo de experimentos. Igualmente, el tamaño de las submuestras es del mismo tamaño que la muestra inicial, dado que se cuenta con escasos recursos.

Después, se entrenó un modelo con cada uno de los dos algoritmos sobre cada una de las 30 submuestras, fijando el valor por defecto de 0.005 del parámetro *finish_threshold.* Luego, se estimaron los parámetros de *precision*, *recall* y *f-measure* para cada experimento como se muestra en las tablas del Error! Reference source not found..

Como se muestra en la **Figura 9**, los resultados del algoritmo *recursive* para las 30 submuestras son superiores a los resultados del algoritmo *Viterbi*. Por consiguiente, se decidió utilizar el algoritmo *recursive* en los experimentos.

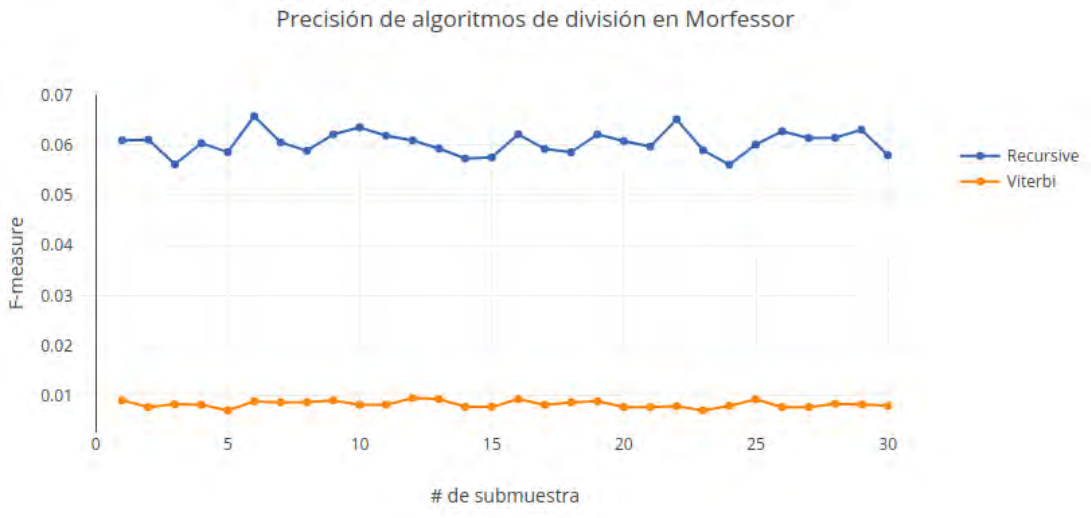


**Figura 9** Gráfico de resultados de *F*-measure de cada algoritmo de segmentación de Morfessor aplicado a 30 submuestras

Por otra parte, para la selección del parámetro *finish_threshold*, se siguió el mismo procedimiento utilizando muestreo con repeticiones. Sobre las mismas 30 submuestras creadas anteriormente, se entrenaron 5 modelos con diferentes valores del parámetro *finish_threshold*, esta vez utilizando el algoritmo *recursive*, seleccionado previamente. En la **Figura 10**, se puede observar las curvas de *precision-recall* obtenidas a partir de los resultados de los 150 experimentos detallados en el Error! Reference source not found..

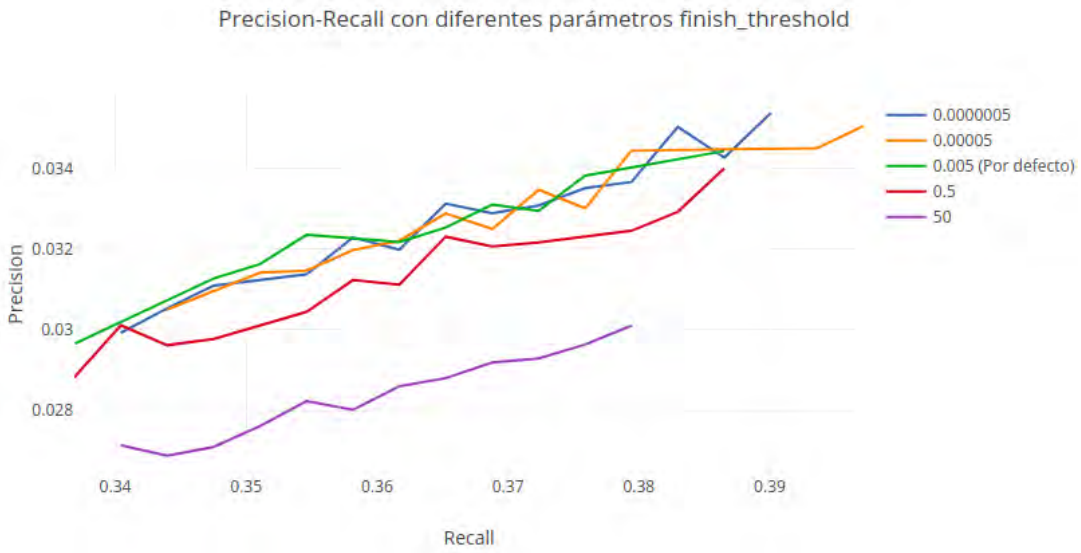


**Figura 10** Gráfico de curvas *precision-recall* utilizando difrentes parámetros *finish_threshold* sobre 30 submuestras

Como se observa en la **Tabla 13**, los valores *precision* y *recall* calculados son casi similares, por lo que optamos mantener el valor 0.005, por defecto para acotar los siguientes experimentos. Cabe resaltar que estos valores son confiables sobre el conjunto original de palabras shiwilu, ya que el error estándar es bastante pequeño.

**Tabla 13** Resultados de *Precision* y *Recall* para diferentes valores de *finish_threshold*

| ***finish_threshold*** | ***Precision*** | ***Recall*** |
|---|---|---|
| 0.000005 | Media: 0.03299<br>Desv. Estándar: 0.00143<br>Error Estándar: 0.00001 | Media: 0.36856<br>Desv. Estándar: 0.01412<br>Error Estándar: 0.00012 |

| *finish_threshold* | *Precision* | *Recall* |
|---|---|---|
| 0.00005 | Media: 0.03273<br>Desv. Estándar: 0.00114<br>Error Estándar: 0.00001 | Media: 0.36619<br>Desv. Estándar: 0.01200<br>Error Estándar: 0.00010 |
| 0.005 | Media: 0.03299<br>Desv. Estándar: 0.00130<br>Error Estándar: 0.00001 | Media: 0.36916<br>Desv. Estándar: 0.01259<br>Error Estándar: 0.00011 |
| 0.5 | Media: 0.03132<br>Desv. Estándar: 0.00121<br>Error Estándar: 0.00001 | Media: 0.36253<br>Desv. Estándar: 0.01326<br>Error Estándar: 0.00011 |
| 50 | Media: 0.02849<br>Desv. Estándar: 0.00098<br>Error Estándar: 0.00001 | Media: 0.36111<br>Desv. Estándar: 0.01137<br>Error Estándar: 0.00010 |

En la **Tabla 14**, se resumen los valores seleccionados para los parámetros de Morfessor que obtuvieron mejores resultados.

**Tabla 14** Valores seleccionados para los parámetros de Morfessor

| Parámetro | Valor seleccionado |
|---|---|
| Tipo de entrenamiento | batch |
| algorithm | recursive |
| finish_threhold | 0.005 |

## 5.2 Optimización de parámetros para Linguistica

Para realizar la segmentación a través de Linguistica, es relevante configurar los parámetros indicados en la **Tabla 15**. En cuanto a los posibles valores a analizar para el parámetro *max_word_types*, se consideró la cantidad total de palabras shiwilu, puesto que los *word types* hacen referencia a la cantidad de palabras únicas en el corpus. En el caso del parámetro *min_stem_length*, se seleccionó el valor de 1, ya que, en los datos extraídos, se ha encontrado palabras formadas por un único caracter. Por otra parte, se decidió analizar los valores 15 y 28 para el parámetro *max_affix_length*, porque son las longitudes máximas encontradas en afijos y palabras, respectivamente.

**Tabla 15** Posibles valores de parámetros de Linguistica

| Parámetro | Valor por defecto | Valores analizados |
|---|---|---|
| max_word_types | 1 000 | 13 377 |
| min_stem_length | 4 | 1 |
| max_affix_length | 4 | 15 |
| | | 28 |
| min_sig_count | 5 | 2 |
| | | 3 |
| | | 4 |
| | | 6 |

Los experimentos practicados en esta sección, se realizaron a través del muestreo con reemplazo, utilizando las mismas 30 submuestras de la sección anterior. En un primer momento, se experimentó con los valores por defecto. Después, los parámetros se fueron variando uno por uno, fijando el resto de parámetros por defecto. Luego, se experimentó con los parámetros que lograron tener mejor resultado.

En el **Anexo D**, se muestran los resultados obtenidos en los experimentos. Se puede observar que la variación del parámetro *max_word_types* no afectó los resultados, por lo tanto, se decidió conservar el valor por defecto. El parámetro *min_stem_length*, obtuvo mejores resultados con un valor de 1. Mientras que *max_affix_length*, obtuvo los mismos resultados para ambos valores seleccionados, además de mejorar al del valor por defecto en cuanto a la métrica *recall*. Sin embargo, se decidió permanecer con el valor por defecto, ya que, si bien identifica menor cantidad de morfemas, estos contienen menos falsos negativos. Por otro lado, *min_sig_count*, obtuvo mejores resultados al ir disminuyen su valor, por tanto, se seleccionó un valor mínimo de 2.

Por último, se observa que la combinación de estos valores seleccionados, brindan mejores resultados frente a los demás experimentos. Se resumen los valores seleccionados para los parámetros de Linguistica en la **Tabla 16**.

**Tabla 16** Valores seleccionados para los parámetros de Linguistica

| Parámetro | Valor seleccionado |
|---|---|
| max_word_types | 1 000 |
| min_stem_length | 1 |
| max_affix_length | 4 |
| min_sig_count | 2 |

## 5.3 Optimización de parámetros para BPE

Para realizar la segmentación morfológica a través de BPE, se debe configurar el parámetro *merge_operations*, el cual indica cuánto iterar para unir las secuencias de caracteres y formar subpalabras. Un valor muy alto de *merge_operations* hace que las subcadenas obtenidas tienda a ser iguales a las palabras originales. Mientras que un valor muy bajo de *merge_operations* hace que las subcadenas tiendan a resultar caracteres.

En la **Tabla 17**, se muestra la lista de valores utilizados en los experimentos para la búsqueda del valor de *merge_operations* óptimo. Estos valores son menores al valor por defecto, dado que los experimentos se realizaron con escasos recursos.

**Tabla 17** Posibles valores de parámetro *merge_operations* para BPE

| Parámetro | Valor por defecto | Valores analizados |
|---|---|---|
| merge_operations | 1 000 | 100<br>200<br>300<br>400<br>500<br>600<br>700<br>800<br>900<br>1 000 |

Se realizaron 30 experimentos para cada uno de los 10 posibles valores de *merge_operations*, haciendo uso de las muestras generadas anteriormente. En el Error! Reference source not found., se resumen los resultados de los experimentos por cada valor de *merge_operations*.

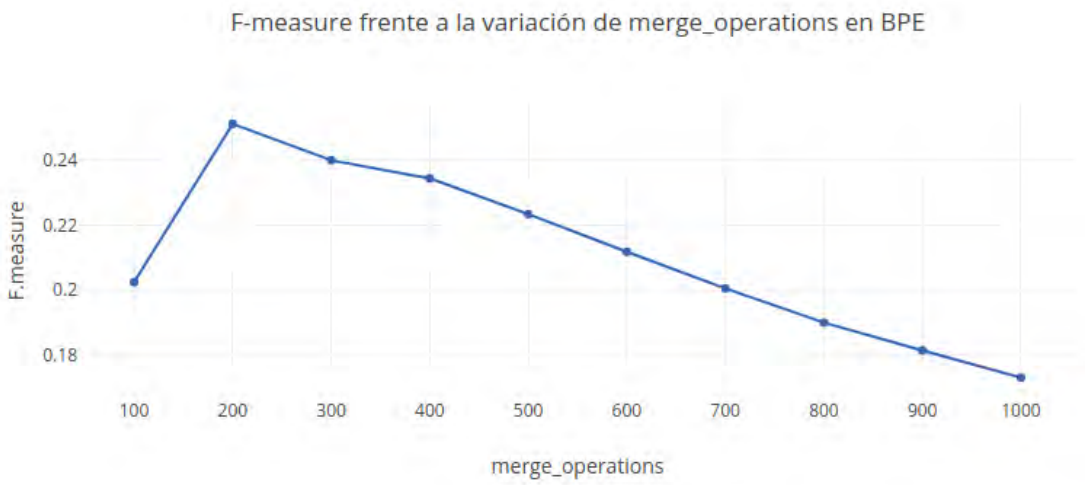


**Figura 11** Resultados de *F-measure* de acuerdo a la variación de *merge_operations* en BPE

Se observa que a partir del valor 200 la precisión empieza a disminuir, tal como se muestra en la **Figura 11**. Por ello, se seleccionó como valor óptimo, como se indica en la **Tabla 18**.

**Tabla 18** Valor seleccionado para *merge_operations* para BPE

| Parámetro | Valor seleccionado |
|---|---|
| merge_operations | 200 |

# Capítulo 6. Análisis morfológico multilingüe

En este capítulo, se abordará el análisis multilingüe para la lengua shiwilu, gracias a recursos en inglés, la cual es una lengua ampliamente estudiada, y con recursos digitalizados. Previo al análisis, se realizó el procesamiento descrito en **Error! Reference source not found.**. Cumpliendo con el objetivo específico 3, se presenta el algoritmo seguido y los morfemas identificados.

## 6.1 Representaciones vectoriales

Para aprender información de una lengua a través de otra lengua, se utilizan dos conjuntos de representaciones vectoriales obtenidas independientemente. Luego, utilizando un diccionario, se entrena una matriz lineal para mapear los vectores de la lengua "origen" a la lengua "objetivo". (Mikolov et al., 2013).

Para poder inducir las categorías gramaticales de la lengua shiwilu, se consideró utilizar representaciones vectoriales de las palabras en inglés y shiwilu. Esto permite inducir las características de una lengua a través de la proyección entre sus espacios vectoriales (Mikolov, Le, & Sutskever, 2013).

### 6.1.1 Creación y alineación de los espacios vectoriales

A través de la librería FastText (Bojanowski, Grave, Joulin & Mikolov, 2017), se creó un espacio de 100 dimensiones para cada lengua, con las representaciones vectoriales de las palabras que tiene más de 10 apariciones en el corpus. Creando modelos con 857 palabras en inglés y 329 en shiwilu.

Tras crear ambos espacios, se alineó el espacio de las palabras en shiwilu respecto al espacio de las palabras en inglés. Para esto, se utilizó el diccionario bilingüe descrito en **Error! Reference source not found.**, el cual permite crear una matriz de transformación y

función de mapeo lineal para obtener una representación de una palabra desconocida en shiwilu en el espacio alineado al espacio inglés.

La función de mapeo permite identificar las características de las palabras shiwilu, a través de su proyección en el espacio inglés, ya que las representaciones vectoriales de palabras similares en diferentes lenguas están relacionadas por una transformación lineal (Mikolov et al., 2013).

### 6.1.2 Validación de alineación de los espacios vectoriales

Para validar la alineación, se calcula la distancia entre palabras relacionadas. Se realizó 2 tipos de validaciones: la primera, entre palabras en shiwilu dentro del mismo espacio y la segunda, entre una palabra shiwilu y su traducción correspondiente en inglés, es decir, entre el espacio inglés y el espacio shiwilu alineado.

En la primera validación, se calculó la distancia entre pronombres en shiwilu (a - b) y entre preposiciones en shiwilu (c – d). En la **Tabla 19** se observa la similitud.

**Tabla 19** Distancia entre pares de palabras shiwilu relacionadas en el espacio vectorial alineado

| Pares de palabras en shiwilu | Distancia entre los pares |
|---|---|
| a) Kua (I) – kenma (you) | 0.999875813359 |
| b) kenmu (we) – nawa (they) | 0.999898857951 |
| c) kawi (near) – malek (because of) | 0.999754416395 |
| d) supinak (supinak) – walek (until) | 0.999898021539 |

Además, como los posibles valores de la distancia varía entre -1 y 1, y en los resultados obtenemos valores cercanos a 1, podemos afirmar que en el espacio de vectores se han agrupado palabras con características similares.

En la segunda validación, se calculó la distancia entre los vectores de los siguientes pares de palabras y sus respectivas traducciones. La distancia entre cada palabra con su

traducción es similar para todos los pares, como se puede observar en los ejemplos de la **Tabla 20**.

**Tabla 20** Distancia entre algunos pares de palabras shiwilu y sus traducciones respecto al espacio vectorial alineado

| Pares de palabras shiwilu - inglés | Distancia entre los pares |
|---|---|
| a) I – kua | 0.305 114 283 711 |
| b) you – kenma | 0.290 657 511 355 |
| c) we – kenmu | 0.298 991 671 943 |
| d) they – nawa | 0.303 290 705 898 |
| e) near – kawi | 0.302 246 250 001 |
| f) because – malek | 0.294 221 614 428 |
| g) after – supinak | 0.298 556 915 390 |
| h) until – walek | 0.298 247 333 929 |

En la **Figura 12**, se muestra la distribución de las distancias entre los pares de palabras shiwilu y sus correspondientes traducciones en inglés del diccionario bilingüe. La distancia entre las palabras es de 0.29995 ± 0.002659.


**Figura 12** Distribución de distancia entre pares de palabras shiwilu e inglés

### 6.1.3 Espacios vectoriales obtenidos

En el **Anexo F**, se pueden observar los dos espacios vectoriales obtenidos, representados en 2 dimensiones. En la **Figura 16**, se muestra el espacio vectorial inglés. Mientras que, en la **Figura 17**, se visualiza el espacio vectorial obtenido con las palabras shiwilu, luego de ser alineado al espacio vectorial inglés gracias a la matriz de transformación.

## 6.2 Identificación de morfemas

Se identificó una lista de posibles morfemas con diferentes tamaños de subcadenas de las palabras, de los cuales solo se consideraron los 500 más frecuentes. En esta última lista, se eliminaron algunos morfemas de acuerdo a la siguiente regla: si un patrón más corto está incluido en un patrón más largo y, además, la relación entre la frecuencia de aparición entre el patrón largo y el corto es mayor a 0.5, entonces el patrón más corto ya no será incluido en la lista (Probst, 2003). Por ejemplo, si tenemos como posibles morfemas:

- -wa' (120 apariciones)
- -mapu'wa' (100 apariciones)

La relación entre las frecuencias de apariciones seria 100/120, esto es mayor que 0.5 por lo tanto ya no se tomaría en cuenta el morfema wa’.

Luego de identificar las características de las palabras shiwilu, estas son agrupadas de acuerdo a características similares; por ejemplo, sustantivos en plural. Se identifica, de la lista de posibles morfemas, cuáles se repiten en este conjunto de palabras, por ejemplo, si un conjunto de palabras siempre tiene un patrón específico, ese morfema representa esa característica (Probst, 2003). Por ejemplo, si tenemos el siguiente grupo de palabras identificado previamente en **Error! Reference source not found.** con la característica "Sustantivos en plural":

- Wilalusa'
- Dudenlusa'
- Wa'danteklusa'
- Wilalunshalusa'
- Ñiñi'walusa'ler

Podemos observar que todas las palabras contienen y comparten la subcadena *lusa'* (identificado previamente como posible morfema), con esto podemos inducir que el sufijo –lusa' indica el plural en los sustantivos.

Cabe mencionar que en este experimento se utilizaron las características mostradas en la **Tabla 21**.

**Tabla 21** Características utilizadas para el análisis multilingüe

| | | | |
|---|---|---|---|
| Adposición | Clusividad | Tipo de sustantivo | Prefijo |
| Adverbio | Tipo de conjunción | Número | Reflexivo |
| Animación | Género | Partícula | Tiempo |
| Aspecto | Modo | Persona | Forma verbal |
| Caso | Sustantivo | Posesivo | Tipo de verbo |

# Capítulo 7. Comparación de enfoques en segmentadores morfológicos

En este capítulo, cumpliendo con el objetivo específico 4, se abordará la experimentación numérica que permite comparar los tres enfoques monolingües y el enfoque multilingüe, para identificar cuál es el que tiene mejores resultados para la segmentación morfológica automática no supervisada en el caso de estudio shiwilu.

## 7.1 Resumen de resultados obtenidos

En la **Tabla 22**, se muestran los promedios de las métricas obtenidas en los enfoques monolingüe y multilingüe, detallados en el **Capítulo 5** y el **Capítulo 6**. Estos resultados se obtuvieron a través del método de muestreo con reemplazo sobre 30 submuestras de 13 377 palabras. Se observa que BPE muestra mejores resultados en *f-measure* que Morfessor, Linguistica y el enfoque multilingüe. A pesar de que Morfessor, obtuvo mayor *recall*, esto solo muestra que identifica menos falsos negativos que BPE.

En cuanto al enfoque multilingüe, este da resultados más exactos pese a identificar menor cantidad de morfemas identificados. Por lo tanto, a pesar de ser pocos, casi la mitad de ellos son morfemas válidos, ya identificados por la lingüista.

**Tabla 22** Comparación de métricas promedio obtenidas

| | **Enfoque** | **Valor** |
|---|---|---|
| **Morfemas Identificados** | Morfessor | 3156.38 ± 45.47 |
| | Linguistica | 2551.86 ± 32.17 |
| | BPE | 201.40 ± 2.28 |
| | Multilingüe | 55.63 ± 3.39 |
| ***Precision*** | Morfessor | 0.03299 ± 0.00130 |
| | Linguistica | 0.02817 ± 0.00092 |
| | BPE | 0.30143 ± 0.00739 |
| | Multilingüe | 0.47098 ± 0.02506 |
| ***Recall*** | Morfessor | 0.36916 ± 0.01259 |
| | Linguistica | 0.25485 ± 0.00755 |
| | BPE | 0.21525 ± 0.00476 |
| | Multilingüe | 0.09129 ± 0.00723 |

| | Enfoque | Valor |
|---|---|---|
| ***F-measure*** | Morfessor | 0.06057 ± 0.00236 |
| | Linguistica | 0.05073 ± 0.00163 |
| | BPE | 0.25114 ± 0.00565 |
| | Multilingüe | 0.15286 ± 0.01105 |

En particular, para el caso de los enfoques BPE y multilingüe, que mostraron mejores resultados, se realizó una comparación de los morfemas identificados en una de las muestras con mayor *f-measure* en ambos enfoques. Los morfemas identificados correctamente tanto por BPE como por el enfoque multilingüe se pueden observar en la **Tabla 23**, mientras que los morfemas en común identificados erróneamente por ambos enfoques, se pueden observar en la

**Tabla 24**.

**Tabla 23** Morfemas identificados correctamente por BPE y enfoque multilingüe

| | | | | |
|---|---|---|---|---|
| a | en | ku | lu’ | pen |
| a’ | etchu | ku’ | lun | pu’ |
| a’su’ | i | la | lusa’ | sa’ |
| an | k | lek | na | tek |
| da | kek | ler | nen | u |
| ek | ker’ | llina’ | ñi | wek |

**Tabla 24** Morfemas identificados erróneamente por BPE y enfoque multilingüe

| | | | | |
|---|---|---|---|---|
| apalli | i’ñi | palli | tapalli | u’ |
| er’ | n | ta’su’ | tulli | un |

En relación con los morfemas identificados erróneamente, es posible que algunos de estos incluyan a otros morfemas pero que al encontrarse juntos en las palabras shiwilu con

mucha frecuencia, hayan sido considerados como un único morfema. Así, el morfema -apalli podría estar conformado por los morfemas -apa y -lli, el morfema -i'ni podría conformarse por -i el morfema -ta'su' podría estar conformado por los morfemas -t y -a'su', el morfema -tapalli podría estar conformado por -t, -apa y -lli y el morfema -tulli podría estar conformado por -t, -u y -lli.

En el caso de los morfemas -er', -n, -palli, -u' y -un, podrían ser parte de otros morfemas los cuales han sido identificados como múltiples morfemas y no como una unidad, todo lo contrario al caso anterior. Por ejemplo, el morfema -er' podría provenir de los morfemas -er'kek o -ker', ya que sí han sido identificados correctamente los morfemas -kek y -k. Entonces en vez de identificar el morfema -er'kek, se podría haber identificado dos morfemas -er' y -kek, tambien en lugar de identificar -ker', pudo haberse identificado dos morfemas -k y -er'. De manera similar, se identificaron incorrectamente los otros los otros cuatro morfemas: -n podría provenir de los morfemas -an, -etchun, -in, -kun, -nan; -palli podría provenir de -apa+-lli; -u' podría provenir de -ku' y -u'ku; mientras que -un podría originarse a partir del morfema -kun.

Finalmente, en el caso del morfema -i'ñi, podría tratarse de algún alomorfo no identificado, ya que solo se ha encontrado el morfema -ñi como reconocido por la lingüista, sin embargo, se puede encontrar este patrón en muchos verbos transitivos, en su mayoría.

### 7.2 Prueba de hipótesis estadística

El proceso de selección de un modelo con mejores resultados, ya sea monolingüe o multilingüe, se realizó a través de la prueba de hipótesis. Este método estadístico cuantifica las respuestas dada una suposición.

Se realizaron los experimentos a través de 30 muestras utilizando muestreo con reemplazo, para los cuales se calculó las métricas *precision*, *recall* y *f-measure*. Por lo tanto,

se tiene una distribución de la precisión para cada modelo. De esta manera, se desea verificar los valores de precisión *f-measure* obtenidos por los modelos, ya que abarcan tanto los valores de *precision* como de *recall*. Las variables utilizadas son:

- X: Valores de *f-measure* obtenidos a través del enfoque Morfessor.
- Y: Valores de *f-measure* obtenidos a través del enfoque Linguistica.
- Z: Valores de *f-measure* obtenidos a través de enfoque BPE.
- W: Valores de *f-measure* obtenidos a través de enfoque multilingüe.

Para poder comparar los resultados de los modelos, primero, se debe conocer la distribución que siguen las variables. Por lo tanto, se utilizó la prueba estadística Kolmogorov-Smirnov para evaluar si cada una de las variables sigue una distribución normal. Se planteó las siguientes hipótesis para cada una:

- $H_0$: La variable sigue una distribución normal.
- $H_1$: La variable no sigue una distribución normal.

En la **Tabla 25**, se muestran los resultados de la prueba Kolmogorov-Smirnov. Como para cada una de las variables, el resultado de la prueba (D) es menor que el valor crítico y el *p-value* es mayor que el nivel de significancia (α), entonces se acepta $H_0$. Por lo tanto, cada variable sigue una distribución normal.

**Tabla 25** Resultados de prueba de normalidad Kolmogorov-Smirnov

| | **X** | **Y** | **Z** | **W** |
|---|---|---|---|---|
| Promedio | 0.060 483 | 0.050 727 | 0.251 142 | 0.152 861 |
| Varianza | 5.59E-06 | 2.67E-06 | 3.19E-05 | 1.22E-04 |
| Muestras | 30 | | | |
| Nivel de significancia (α) | 0.05 | | | |
| Valor crítico | 0.2417 | | | |
| D | 0.068 11 | 0.126 55 | 0.073 54 | 0.117 |

| | | | | |
|---|---|---|---|---|
| *p-value* | 0.997 37 | 0.676 05 | 0.992 99 | 0.763 05 |

Ya que todas las variables presentan una distribución normal, el siguiente paso es conocer si las muestras provienen de poblaciones con la misma varianza. Para esto, se utilizó la prueba de homocedasticidad de Barlett. Se planteó las siguientes hipótesis:

- $H_0$: Las distribuciones X, Y, Z y W presentan varianzas similares.
- $H_1$: Las distribuciones X, Y, Z y W no presentan varianzas similares.

En la **Tabla 26**, se muestran los resultados de la prueba de Barlett. Como el *p-value* es mucho menor que el nivel de significancia (α), se rechaza $H_0$. Por lo tanto, las distribuciones no presentan varianzas similares.

**Tabla 26** Resultados de prueba de homocedasticidad Barlett

| | **Resultados** |
|---|---|
| Nivel de significancia (α) | 0.05 |
| T | 110.079 400 |
| *p-value* | 1.19E-23 |

Debido a que las variables presentan varianzas diferentes, se utilizó el método Welch ANOVA para la comparación de medias. Se planteó las siguientes hipótesis:

- $H_0$: Todas las medias de las distribuciones son iguales y no tienen diferencias significativas.
- $H_1$: Por lo menos una de las medias de las distribuciones es diferente y tienen diferencias significativas.

En la **Tabla 27**, se muestran los resultados de la prueba Welch ANOVA. Puesto que el *p-value* es menor que el nivel de significancia (α), se rechaza $H_0$.

**Tabla 27** Resultados de prueba de comparación de medias Welch ANOVA

| | **Resultados** |
|---|---|
| Nivel de significancia (α) | 0.05 |
| F | 1 857.731 35 |

| | |
|---|---|
| *p-value* | 1.95E-30 |

Finalmente, para identificar el modelo con mejores resultados, se realizó una prueba *post hoc*. La prueba de Hsu de múltiples comparaciones con el mejor (MCB) se utilizó para comparar las métricas *f-measure* con la "mejor" media, en este caso, la media de *f-measure* del modelo BPE. En la **Figura 13**, se muestran los intervalos de confianza obtenidos.

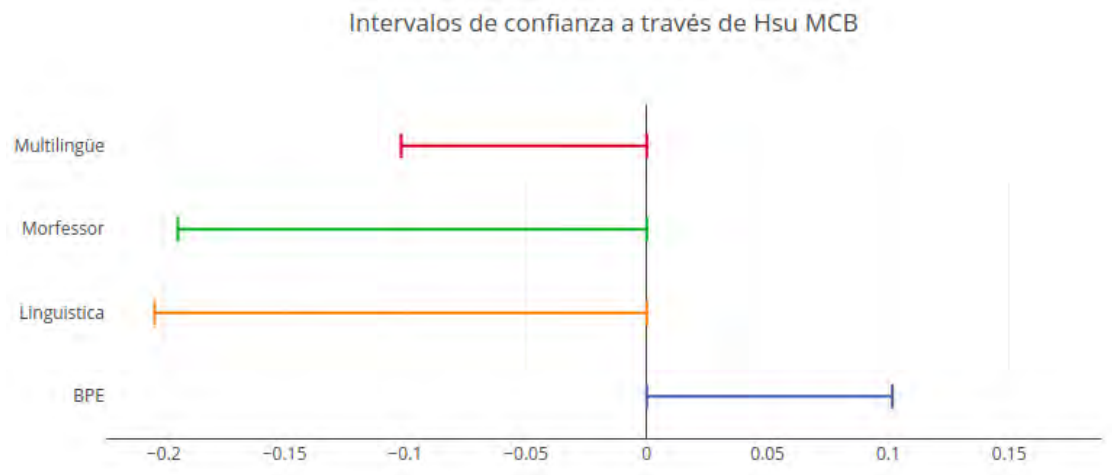


**Figura 13** Gráfico de intervalos de confianza en comparaciones múltiples con el mejor (MCB)

Visto que los intervalos de confianza para Morfessor, Linguistica y el enfoque multilingüe están por debajo de cero, estos son significativamente peor que BPE. En conclusión, el modelo BPE es significativamente mejor, dado que el intervalo de confianza de BPE está por encima de cero.

# Capítulo 8. Centralización de recursos en la web

En este capítulo, se detalla cómo se integra el enfoque más preciso con un aplicativo web que permitirá centralizar los recursos shiwilu, tal como se observa en el diagrama de despliegue de la **Figura 14**.

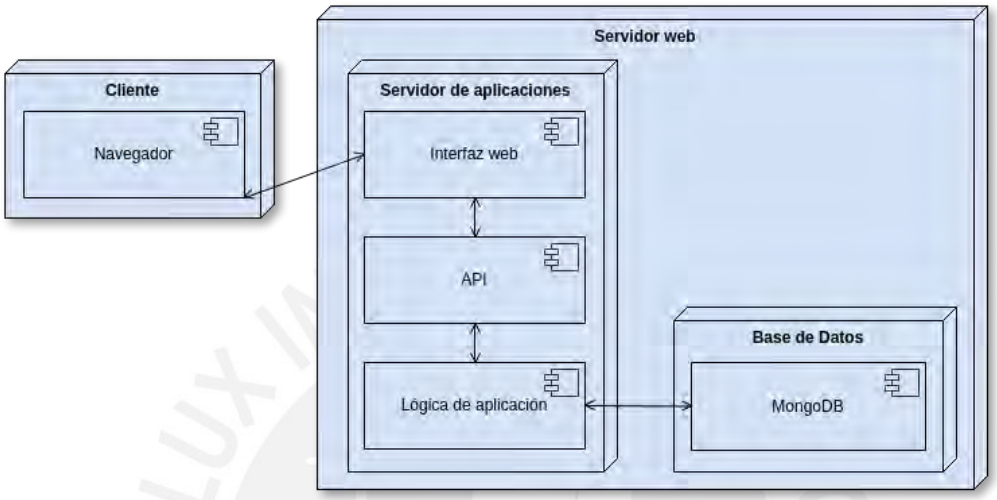


**Figura 14** Diagrama de despliegue de aplicativo web

## 8.1 Interfaz web

Para poder aprovechar los recursos digitales y esta herramienta automática de análisis, se ha creado una interfaz donde se podrá realizar consultas por palabra en shiwilu. La consulta permitirá visualizar la segmentación morfológica con sus características, definiciones de la palabra en inglés y/o español (según esté disponible), y ejemplos en shiwilu con sus respectivas traducciones al inglés y/o español (según esté disponible).

En la **Figura 15**, se muestra la segmentación de la palabra *ñiñi'waluna'* (perros). Donde *ñiñi'wa* es la raíz y *luna'* es el sufijo que indica el plural. Al lado izquierdo, se muestra la definición de *ñiñi'wa* (perro) y algunos ejemplos, con sus respectivas traducciones.


**Figura 15** Ejemplo de resultado de consulta en el aplicativo web

## 8.2 API

Se ha creado un API con Django a través del cual se pueden realizar múltiples consultas, tomando como entrada una palabra en shiwilu:

- **Segmentador**. Segmenta una palabra a través del método monolingüe gracias a la herramienta BPE, la cual tuvo mejor desempeño frente a los otros análisis.
- **Consulta de definiciones**. Devuelve una o más definiciones de la palabra shiwilu en español y/o inglés de acuerdo a disponibilidad en la base de datos.
- **Consulta de ejemplos**. Devuelve uno o más ejemplos por cada acepción de la palabra shiwilu. Los ejemplos están en shiwilu con sus respectivas traducciones al inglés y/o español. Los ejemplos dependen de la disponibilidad en la base de datos.

# Capítulo 9. Conclusiones y trabajos futuros

## 9.1 Conclusiones

El presente proyecto tomó como caso de estudio la lengua shiwilu que, pese a sus escasos recursos, se logró cumplir con el objetivo específico 1, lo que involucró un extenso preprocesamiento, puesto que se desconocía a detalle la información recibida inicialmente (la base de un diccionario shiwilu), además, de la falta de estandarización en los datos, al encontrarse aun pequeñas anotaciones que servían como ayuda en el trabajo de campo del levantamiento de información lingüístico. No obstante, se logró con éxito normalizar y almacenar los datos de las palabras shiwilu para su posterior aprovechamiento.

En cuanto al objetivo específico 2 y el análisis monolingüe, se optimizaron parámetros para tres algoritmos de segmentación, a través de la búsqueda de los valores más altos para las métricas de *precision* y *recall*, los cuales mostraban resultados más exactos y correctos. Sin embargo, al no poder diferenciar afijos o raíces entre los morfemas identificados, para poder ser validados con la lista de afijos brindados por la lingüista, hizo que los valores de las métricas sean muy pequeños, pero, aun así, estadísticamente comparables entre sí.

Por otra parte, al tratarse de una lengua aglutinante, la complejidad aumentaba por su gran riqueza morfológica, pudiendo encontrar múltiples morfemas en una sola palabra, haciendo que las cadenas de palabras sean bastante largas y extensas en su procesamiento, tal como sucede en el estudio realizado por Kumar, Padró & Oliver en el 2015, indicado en el estado del arte.

Por lo que se refiere al objetivo específico 3, para el enfoque multilingüe, la creación del diccionario bilingüe también tuvo un grado de complejidad, ya que los recursos eran escasos para este tipo de construcción. No obstante, se pudo construir gracias al diccionario base, algunos préstamos entre lenguas, y nombres propios que no varían entre lenguas. Por

otra parte, utilizando el idioma inglés, al igual que Baumann y Pierrehumbert - indicado en el estado del arte, se obtuvo una cantidad similar de morfemas a los que encontraron en la lengua zulú.

Para poder aprovechar los escasos recursos y entrenar con éxito los modelos, se utilizó el método de *bootstrapping with replacement*, el cual es una técnica estadística que permite estimar métricas o hacer predicciones sobre la población a partir de submuestras. Esto permitió estimar la eficiencia de los modelos, promediándolos a través de 30 submuestras.

Dentro de este marco, se cumple el objetivo específico 4, dado que diferentes pruebas estadísticas permitieron verificar que el enfoque con mejores resultados fue BPE, ya que el modelo da resultados más exactos, lo cual se ve reflejado en el valor de *precision*. Finalmente, comparando *precision* y *recall* a través de la métrica *F-measure*, BPE obtuvo mejores resultados que Morfessor, Linguistica y el enfoque multilingüe, a pesar de identificar menor cantidad de posibles morfemas que los otros dos enfoques monolingües.

## 9.2 Trabajos futuros

Como trabajos futuros, podrían mejorarse los resultados obtenidos con el enfoque monolingüe, a través de un modelo semi supervisado que pueda tomar inicialmente información de palabras ya segmentadas morfológicamente.

Para el enfoque multilingüe, podrían mejorarse los resultados, a través de un diccionario bilingüe más amplio, lo cual permitiría una alineación más precisa de los espacios vectoriales. En cuanto a la identificación de características, si se contara con mejores modelos pre entrenados de información morfológica en inglés, podría beneficiar a la identificación de características en shiwilu.

**Referencias**

Abell, M. L., Braselton, J. P., Rafter, J. A., & Rafter, J. A. (1999). *Statistics with mathematica (Vol. 1)*. Academic Press.

Ablimit, M., Kawahara, T., Pattar, A., & Hamdulla, A. (2016). Stem-affix based Uyghur morphological analyzer. International Journal of Future Generation Communication and Networking, 9(2), 59-72. Alpaydin, E. (2010). *Introduction to machine learning*. MIT press.

Acosta, S., Natalia, K., Huamancayo Curi, E., Mori Clement, M., & Carbajal Solis, V. (2013). *Documento nacional de lenguas originarias del Perú*.

Ács, J., & Velkey, G. Comparing word segmentation algorithms.

Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

Baumann, P., & Pierrehumbert, J. B. (2014). Using Resource-Rich Languages to Improve Morphological Analysis of Under-Resourced Languages. In *LREC* (pp. 3355-3359).

Bender, E. M. (2013). Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies*, 6(3), 1-184.

Bhat, S. (2012). Morpheme segmentation for kannada standing on the shoulder of giants. In 24th International Conference on Computational Linguistics (p. 79).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.Borin, L. (2009). Linguistic diversity in the information society. In *Proceedings of the saltmil 2009 workshop on information retrieval and information extraction for less resourced languages* (pp. 1-7).

Brownlee, J. (2018). *Statistical methods for machine learning: Discover how to transform data into knowledge with Python*. Machine Learning Mastery.

Campbell, L., & Grondona, V. (Eds.). (2012). *The indigenous languages of South America: A comprehensive guide* (Vol. 2). Walter de Gruyter.

Cerrón-Palomino, R. (1994). *Quechumara: estructuras paralelas de las lenguas quechua y aimara* (Vol. 42). La Paz: Centro de Investigación y Promoción del Campesinado.

Creutz, M., & Lagus, K. (2005). *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0.* Helsinki University of Technology.

Crowley, T. (2007). *Field linguistics: A beginner's guide.* OUP Oxford.

Crystal, D. (2011). *Dictionary of linguistics and phonetics (Vol. 30)*. John Wiley & Sons.

De Muth, J. E. (2014). *Basic statistics and pharmaceutical statistical applications*. CRC Press.

Fase, W., Jaspaert, K., & Kroon, S. (1992). *Maintenance and Loss of Minority Languages*. LEARNING MATTERS.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37.

Forcada, M. (2006). Open source machine translation: an opportunity for minor languages. In *Proceedings of the Workshop "Strategies for developing machine translation for minority languages", LREC* (Vol. 6, pp. 1-6).

Hammarström, H., & Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics*, 37(2), 309-350.

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier

Haspelmath, M. (2002). *Understanding Morphology*. London: Arnold.

Haspelmath, M., & Sims, A. (2010). *Understanding Morphology*. Oxford University Press.

Kumar, A., Padró, L., & Oliver, A. (2015, October). Unsupervised learning of agglutinated morphology using nested Pitman-Yor process based morpheme induction algorithm. In *Asian Language Processing (IALP)*, 2015 International Conference on (pp. 45-48). IEEE.

Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999, February). Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop* (pp. 249-252).

Management Association, I. R. (2011). *Machine Learning: Concepts, Methodologies, Tools and Applications*. Information Science Reference.

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

Marques, M. (2001). *Estadística Básica: un enfoque no paramétrico*. Ciudad de México: Universidad Nacional Autónoma de México.

Miangah, T. M. (2012). The Role of Large Monolingual Corpora. In *Improving Machine Translatio*n *Quality*, 31.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation.

Mongodb.com (2012). MongoDB Architecture. Recuperado de https://www.mongodb.com/mongodb-architecture

Moon, T. (2008). Minimally supervised induction of morphology through bitexts.

O'Horan, H., Berzak, Y., Vulić, I., Reichart, R., & Korhonen, A. (2016). Survey on the use of typological information in natural language processing.

Probst, K. (2003). Using smart bilingual projection to feature-tag a monolingual dictionary. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL* 2003-Volume 4 (pp. 103-110). Association for Computational Linguistics.

Python Documentation (2012). Difflib - Helpers for computing deltas. Recuperado de https://docs.python.org/2/library/difflib.html

Real Academia Española. (2016). *Gramática básica de la lengua española*. Grupo Planeta.

Richards, J. C., & Schmidt, R. W. (2013). *Longman dictionary of language teaching and applied linguistics*. Routledge.

Samarin, W. J. (1967). *Field linguistics: A guide to linguistic field work.* Holt, Rinehart and Winston.

Sánchez, M. G. P. (2014). *Cuestiones de morfología español*a. Editorial Universitaria Ramón Areces.

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

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units.

Shalonova, K., & Golénia, B. (2010, August). Weakly supervised morphology learning for agglutinating languages using small training sets. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 976-983). Association for Computational Linguistics.

Shalonova, K., Golénia, B., & Flach, P. (2009). *Towards learning morphology for under-resourced fusional and agglutinating languages. IEEE Transactions on Audio, Speech, and Language Processing*, 17(5), 956-965.

Štekauer, P., Valera, S., & Kőrtvélyessy, L. (2012). *Word-Formation in the World's Languages: A Typological Survey*. Cambridge University Press.

Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. Apress

Valenzuela, P. M. (2008). SHIWILU LA'LA'MAPU'WA'Los desafíos de una lengua en vías de desaparición en la Amazonía Peruana. In *First Conference on Ethnicity, Race, and Indigenous Peoples in Latin America & the Caribbean UCSD* (pp. 22-24).

Weerahandi, S. (2013). *Exact statistical methods for data analysis*. Springer Science & Business Media.

Wheeldon, L. (Ed.). (2002). *Aspects of language production*. Psychology Press.

Winter, Werner. 1993. Some conditions for the survival of small languages. In Ernst Hakon Jahr, editor, Language conflict and language planning. Mouton de Gruyter, Berlin, pages 299–314.

## Anexo A. Resultados de experimentos basados en posibles combinaciones de valores para parámetros de Morfessor

| Número de experimento | Tipo de entrenamiento | Parámetros | | Resultados | | |
|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_threshold* | *precision* | *recall* | *F-measure* |
| 1 | batch | recursive | 0.0000005 | 0.0464066 | 0.4007092 | 0.0831800 |
| 2 | batch | recursive | 0.000005 | 0.0431596 | 0.3758865 | 0.0774288 |
| 3 | batch | recursive | 0.00005 | 0.0466278 | 0.3971631 | 0.0834575 |
| 4 | batch | recursive | 0.0005 | 0.0461601 | 0.4007092 | 0.0827839 |
| 5 | batch | recursive | 0.005 | 0.0441653 | 0.3865248 | 0.0792727 |
| 6 | batch | recursive | 0.05 | 0.0447220 | 0.3936170 | 0.0803184 |
| 7 | batch | recursive | 0.5 | 0.0456154 | 0.4113475 | 0.0821239 |
| 8 | batch | recursive | 5 | 0.0424041 | 0.4078014 | 0.0768203 |
| 9 | batch | recursive | 50 | 0.0421957 | 0.4007092 | 0.0763514 |
| 10 | batch | recursive | 500 | 0.0423856 | 0.4007092 | 0.0766621 |
| 11 | batch | viterbi | 0.0000005 | 0.0043358 | 0.2056738 | 0.0084926 |
| 12 | batch | viterbi | 0.000005 | 0.0043358 | 0.2056738 | 0.0084926 |
| 13 | batch | viterbi | 0.00005 | 0.0043358 | 0.2056738 | 0.0084926 |
| 14 | batch | viterbi | 0.0005 | 0.0043358 | 0.2056738 | 0.0084926 |
| 15 | batch | viterbi | 0.005 | 0.0043358 | 0.2056738 | 0.0084926 |
| 16 | batch | viterbi | 0.05 | 0.0043358 | 0.2056738 | 0.0084926 |
| 17 | batch | viterbi | 0.5 | 0.0043358 | 0.2056738 | 0.0084926 |
| 18 | batch | viterbi | 5 | 0.0043358 | 0.2056738 | 0.0084926 |
| 19 | batch | viterbi | 50 | 0.0043358 | 0.2056738 | 0.0084926 |
| 20 | batch | viterbi | 500 | 0.0043358 | 0.2056738 | 0.0084926 |

## Anexo B. Resultados de experimentos para descarte de algoritmo de división de Morfessor

| Número de submuestra | Tipo de entrenamiento | Parámetros | | Resultados | | | |
|---|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_threshold* | Total de morfemas identificados | *precision* | *recall* | *F-measure* |
| 1 | batch | recursive | 0.005 | 3164 | 0.0331858 | 0.3723404 | 0.0609402 |
| 2 | batch | recursive | 0.005 | 3157 | 0.0332594 | 0.3723404 | 0.0610643 |
| 3 | batch | recursive | 0.005 | 3208 | 0.0305486 | 0.3475177 | 0.0561605 |
| 4 | batch | recursive | 0.005 | 3130 | 0.0329073 | 0.3652482 | 0.0603751 |
| 5 | batch | recursive | 0.005 | 3199 | 0.0318850 | 0.3617021 | 0.0586038 |
| 6 | batch | recursive | 0.005 | 3125 | 0.0358400 | 0.3971631 | 0.0657470 |
| 7 | batch | recursive | 0.005 | 3187 | 0.0329463 | 0.3723404 | 0.0605362 |
| 8 | batch | recursive | 0.005 | 3217 | 0.0320174 | 0.3652482 | 0.0588740 |
| 9 | batch | recursive | 0.005 | 3098 | 0.0338928 | 0.3723404 | 0.0621302 |
| 10 | batch | recursive | 0.005 | 3149 | 0.0346142 | 0.3865248 | 0.0635383 |
| 11 | batch | recursive | 0.005 | 3112 | 0.0337404 | 0.3723404 | 0.0618739 |
| 12 | batch | recursive | 0.005 | 3131 | 0.0332162 | 0.3687943 | 0.0609435 |
| 13 | batch | recursive | 0.005 | 3158 | 0.0322989 | 0.3617021 | 0.0593023 |
| 14 | batch | recursive | 0.005 | 3242 | 0.0311536 | 0.3581560 | 0.0573212 |
| 15 | batch | recursive | 0.005 | 3194 | 0.0313087 | 0.3546099 | 0.0575374 |
| 16 | batch | recursive | 0.005 | 3064 | 0.0339426 | 0.3687943 | 0.0621638 |
| 17 | batch | recursive | 0.005 | 3196 | 0.0322278 | 0.3652482 | 0.0592294 |
| 18 | batch | recursive | 0.005 | 3099 | 0.0319458 | 0.3510638 | 0.0585626 |
| 19 | batch | recursive | 0.005 | 3226 | 0.0337880 | 0.3865248 | 0.0621437 |
| 20 | batch | recursive | 0.005 | 3139 | 0.0331316 | 0.3687943 | 0.0608009 |
| 21 | batch | recursive | 0.005 | 3169 | 0.0325024 | 0.3652482 | 0.0596928 |
| 22 | batch | recursive | 0.005 | 3096 | 0.0355297 | 0.3900709 | 0.0651273 |
| 23 | batch | recursive | 0.005 | 3177 | 0.0321058 | 0.3617021 | 0.0589766 |
| 24 | batch | recursive | 0.005 | 3177 | 0.0305319 | 0.3439716 | 0.0560856 |
| 25 | batch | recursive | 0.005 | 3079 | 0.0328029 | 0.3581560 | 0.0601012 |
| 26 | batch | recursive | 0.005 | 3128 | 0.0342072 | 0.3794326 | 0.0627566 |
| 27 | batch | recursive | 0.005 | 3170 | 0.0334385 | 0.3758865 | 0.0614137 |
| 28 | batch | recursive | 0.005 | 3200 | 0.0334375 | 0.3794326 | 0.0614589 |
| 29 | batch | recursive | 0.005 | 3144 | 0.0343511 | 0.3829787 | 0.0630473 |
| 30 | batch | recursive | 0.005 | 3202 | 0.0315428 | 0.3581560 | 0.0579793 |

| Número de submuestra | Tipo de entrenamiento | Parámetros | | Resultados | | | |
|---|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_threshold* | Total de morfemas identificados | *precision* | *recall* | *F-measure* |
| 1 | batch | viterbi | 0.005 | 8491 | 0.0047109 | 0.1418440 | 0.0091189 |
| 2 | batch | viterbi | 0.005 | 8475 | 0.0040118 | 0.1205674 | 0.0077652 |
| 3 | batch | viterbi | 0.005 | 8360 | 0.0043062 | 0.1276596 | 0.0083314 |
| 4 | batch | viterbi | 0.005 | 8476 | 0.0042473 | 0.1276596 | 0.0082211 |
| 5 | batch | viterbi | 0.005 | 8453 | 0.0036673 | 0.1099291 | 0.0070979 |
| 6 | batch | viterbi | 0.005 | 8465 | 0.0046072 | 0.1382979 | 0.0089173 |
| 7 | batch | viterbi | 0.005 | 8435 | 0.0045050 | 0.1347518 | 0.0087186 |
| 8 | batch | viterbi | 0.005 | 8408 | 0.0045195 | 0.1347518 | 0.0087457 |
| 9 | batch | viterbi | 0.005 | 8491 | 0.0047109 | 0.1418440 | 0.0091189 |
| 10 | batch | viterbi | 0.005 | 8455 | 0.0042578 | 0.1276596 | 0.0082408 |
| 11 | batch | viterbi | 0.005 | 8444 | 0.0042634 | 0.1276596 | 0.0082512 |
| 12 | batch | viterbi | 0.005 | 8481 | 0.0049522 | 0.1489362 | 0.0095858 |
| 13 | batch | viterbi | 0.005 | 8460 | 0.0048463 | 0.1453901 | 0.0093800 |
| 14 | batch | viterbi | 0.005 | 8440 | 0.0040284 | 0.1205674 | 0.0077964 |
| 15 | batch | viterbi | 0.005 | 8408 | 0.0040438 | 0.1205674 | 0.0078251 |
| 16 | batch | viterbi | 0.005 | 8470 | 0.0048406 | 0.1453901 | 0.0093693 |
| 17 | batch | viterbi | 0.005 | 8463 | 0.0042538 | 0.1276596 | 0.0082333 |
| 18 | batch | viterbi | 0.005 | 8433 | 0.0045061 | 0.1347518 | 0.0087206 |
| 19 | batch | viterbi | 0.005 | 8458 | 0.0046110 | 0.1382979 | 0.0089245 |
| 20 | batch | viterbi | 0.005 | 8475 | 0.0040118 | 0.1205674 | 0.0077652 |
| 21 | batch | viterbi | 0.005 | 8456 | 0.1205674 | 0.1205674 | 0.0077821 |
| 22 | batch | viterbi | 0.005 | 8535 | 0.0041008 | 0.1241135 | 0.0079392 |
| 23 | batch | viterbi | 0.005 | 8462 | 0.0036634 | 0.1099291 | 0.0070906 |
| 24 | batch | viterbi | 0.005 | 8445 | 0.0041445 | 0.1241135 | 0.0080211 |
| 25 | batch | viterbi | 0.005 | 8525 | 0.0048094 | 0.1453901 | 0.0093108 |
| 26 | batch | viterbi | 0.005 | 8471 | 0.0040137 | 0.1205674 | 0.0077688 |
| 27 | batch | viterbi | 0.005 | 8459 | 0.0040194 | 0.1205674 | 0.0077794 |
| 28 | batch | viterbi | 0.005 | 8482 | 0.0043622 | 0.1312057 | 0.0084436 |
| 29 | batch | viterbi | 0.005 | 8412 | 0.0042796 | 0.1276596 | 0.0082816 |
| 30 | batch | viterbi | 0.005 | 8475 | 0.0041298 | 0.1241135 | 0.0079936 |

## Anexo C. Resultados de experimentos para selección del valor del parámetro *finish_threshold* de Morfessor

| Número de submuestra | Tipo de entrenamiento | Parámetros | | Resultados | | | |
|---|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_ threshold* | Total de morfemas identificados | *precision* | *recall* | *F-measure* |
| 1 | batch | recursive | 0.0000005 | 3189 | 0.0319849 | 0.3617021 | 0.0587727 |
| | | | 0.00005 | 3174 | 0.0318210 | 0.3581560 | 0.0584491 |
| | | | 0.5 | 3308 | 0.0296252 | 0.3475177 | 0.0545961 |
| | | | 50 | 3576 | 0.0288031 | 0.3652482 | 0.0533955 |
| 2 | batch | recursive | 0.0000005 | 3209 | 0.0339670 | 0.3865248 | 0.0624463 |
| | | | 0.00005 | 3223 | 0.0322681 | 0.3687943 | 0.0593438 |
| | | | 0.5 | 3316 | 0.0307600 | 0.3617021 | 0.0566982 |
| | | | 50 | 3601 | 0.0291586 | 0.3723404 | 0.0540819 |
| 3 | batch | recursive | 0.0000005 | 3208 | 0.0299252 | 0.3404255 | 0.0550143 |
| | | | 0.00005 | 3219 | 0.0310655 | 0.3546099 | 0.0571265 |
| | | | 0.5 | 3281 | 0.0295642 | 0.3439716 | 0.0544485 |
| | | | 50 | 3638 | 0.0274876 | 0.3546099 | 0.0510204 |
| 4 | batch | recursive | 0.0000005 | 3120 | 0.0349359 | 0.3865248 | 0.0640800 |
| | | | 0.00005 | 3130 | 0.0329073 | 0.3652482 | 0.0603751 |
| | | | 0.5 | 3241 | 0.0311632 | 0.3581560 | 0.0573375 |
| | | | 50 | 3609 | 0.0279856 | 0.3581560 | 0.0519147 |
| 5 | batch | recursive | 0.0000005 | 3147 | 0.0349539 | 0.3900709 | 0.0641586 |
| | | | 0.00005 | 3223 | 0.0307167 | 0.3510638 | 0.0564907 |
| | | | 0.5 | 3270 | 0.0299694 | 0.3475177 | 0.0551802 |
| | | | 50 | 3609 | 0.0268773 | 0.3439716 | 0.0498586 |
| 6 | batch | recursive | 0.0000005 | 3160 | 0.0338608 | 0.3794326 | 0.0621732 |
| | | | 0.00005 | 3150 | 0.0333333 | 0.3723404 | 0.0611888 |
| | | | 0.5 | 3297 | 0.0324537 | 0.3794326 | 0.0597932 |
| | | | 50 | 3571 | 0.0280034 | 0.3546099 | 0.0519076 |
| 7 | batch | recursive | 0.0000005 | 3183 | 0.0333019 | 0.3758865 | 0.0611833 |
| | | | 0.00005 | 3195 | 0.0350548 | 0.3971631 | 0.0644234 |
| | | | 0.5 | 3280 | 0.0329268 | 0.3829787 | 0.0606401 |
| | | | 50 | 3596 | 0.0283648 | 0.3617021 | 0.0526044 |
| 8 | batch | recursive | 0.0000005 | 3207 | 0.0324291 | 0.3687943 | 0.0596159 |
| | | | 0.00005 | 3214 | 0.0329807 | 0.3758865 | 0.0606407 |
| | | | 0.5 | 3330 | 0.0321321 | 0.3794326 | 0.0592470 |
| | | | 50 | 3577 | 0.0296338 | 0.3758865 | 0.0549365 |
| 9 | batch | recursive | 0.0000005 | 3101 | 0.0348275 | 0.3829787 | 0.0638487 |
| | | | 0.00005 | 3046 | 0.0344714 | 0.3723404 | 0.0631010 |
| | | | 0.5 | 3292 | 0.0297691 | 0.3475177 | 0.0548405 |
| | | | 50 | 3527 | 0.0294868 | 0.3687943 | 0.0546075 |
| 10 | batch | recursive | 0.0000005 | 3144 | 0.0337150 | 0.3758865 | 0.0618797 |
| | | | 0.00005 | 3178 | 0.0327250 | 0.3687943 | 0.0601156 |

| Número de submuestra | Tipo de entrenamiento | Parámetros | | Resultados | | | |
|---|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_ threshold* | Total de morfemas identificados | *precision* | *recall* | *F-measure* |
| | | | 0.5 | 3225 | 0.0322481 | 0.3687943 | 0.0593100 |
| | | | 50 | 3585 | 0.0276151 | 0.3510638 | 0.0512025 |
| 11 | batch | recursive | 0.0000005 | 3127 | 0.0322993 | 0.3581560 | 0.0592549 |
| | | | 0.00005 | 3112 | 0.0337404 | 0.3723404 | 0.0618739 |
| | | | 0.5 | 3206 | 0.0324392 | 0.3687943 | 0.0596330 |
| | | | 50 | 3531 | 0.0283206 | 0.3546099 | 0.0524521 |
| 12 | batch | recursive | 0.0000005 | 3111 | 0.0331083 | 0.3652482 | 0.0607132 |
| | | | 0.00005 | 3137 | 0.0334715 | 0.3723404 | 0.0614215 |
| | | | 0.5 | 3220 | 0.0326087 | 0.3723404 | 0.0599657 |
| | | | 50 | 3590 | 0.0281337 | 0.3581560 | 0.0521694 |
| 13 | batch | recursive | 0.0000005 | 3128 | 0.0322890 | 0.3581560 | 0.0592375 |
| | | | 0.00005 | 3134 | 0.0328653 | 0.3652482 | 0.0603044 |
| | | | 0.5 | 3188 | 0.0323087 | 0.3652482 | 0.0593660 |
| | | | 50 | 3537 | 0.0288380 | 0.3617021 | 0.0534171 |
| 14 | batch | recursive | 0.0000005 | 3177 | 0.0305319 | 0.3439716 | 0.0560856 |
| | | | 0.00005 | 3180 | 0.0305031 | 0.3439716 | 0.0560370 |
| | | | 0.5 | 3296 | 0.0288228 | 0.3368794 | 0.0531023 |
| | | | 50 | 3605 | 0.0280166 | 0.3581560 | 0.0519681 |
| 15 | batch | recursive | 0.0000005 | 3123 | 0.0313801 | 0.3475177 | 0.0575624 |
| | | | 0.00005 | 3151 | 0.0314186 | 0.3510638 | 0.0576755 |
| | | | 0.5 | 3262 | 0.0318823 | 0.3687943 | 0.0586907 |
| | | | 50 | 3636 | 0.0266777 | 0.3439716 | 0.0495151 |
| 16 | batch | recursive | 0.0000005 | 3091 | 0.0339696 | 0.3723404 | 0.0622591 |
| | | | 0.00005 | 3086 | 0.0333765 | 0.3652482 | 0.0611639 |
| | | | 0.5 | 3188 | 0.0301129 | 0.3404255 | 0.0553314 |
| | | | 50 | 3538 | 0.0271340 | 0.3404255 | 0.0502618 |
| 17 | batch | recursive | 0.0000005 | 3192 | 0.0328947 | 0.3723404 | 0.0604491 |
| | | | 0.00005 | 3200 | 0.0328125 | 0.3723404 | 0.0603102 |
| | | | 0.5 | 3298 | 0.0318375 | 0.3723404 | 0.0586592 |
| | | | 50 | 3575 | 0.0293706 | 0.3723404 | 0.0544465 |
| 18 | batch | recursive | 0.0000005 | 3107 | 0.0331509 | 0.3652482 | 0.0607849 |
| | | | 0.00005 | 3125 | 0.0316800 | 0.3510638 | 0.0581156 |
| | | | 0.5 | 3269 | 0.0296727 | 0.3439716 | 0.0546325 |
| | | | 50 | 3542 | 0.0282326 | 0.3546099 | 0.0523013 |
| 19 | batch | recursive | 0.0000005 | 3197 | 0.0334689 | 0.3794326 | 0.0615119 |
| | | | 0.00005 | 3217 | 0.0345042 | 0.3936170 | 0.0634467 |
| | | | 0.5 | 3292 | 0.0325030 | 0.3794326 | 0.0598769 |
| | | | 50 | 3570 | 0.0299720 | 0.3794326 | 0.0555556 |
| 20 | batch | recursive | 0.0000005 | 3187 | 0.0313775 | 0.3546099 | 0.0576535 |
| | | | 0.00005 | 3167 | 0.0322071 | 0.3617021 | 0.0591476 |
| | | | 0.5 | 3273 | 0.0305530 | 0.3546099 | 0.0562588 |

| Número de submuestra | Tipo de entrenamiento | Parámetros | | Resultados | | | |
|---|---|---|---|---|---|---|---|
| | | *algorithm* | *finish_ threshold* | Total de morfemas identificados | *precision* | *recall* | *F-measure* |
| | | | 50 | 3573 | 0.0291072 | 0.3687943 | 0.0539559 |
| 21 | batch | recursive | 0.0000005 | 3181 | 0.0308079 | 0.3475177 | 0.0565983 |
| | | | 0.00005 | 3127 | 0.0338983 | 0.3758865 | 0.0621883 |
| | | | 0.5 | 3321 | 0.0313159 | 0.3687943 | 0.0577297 |
| | | | 50 | 3581 | 0.0296007 | 0.3758865 | 0.0548796 |
| 22 | batch | recursive | 0.0000005 | 3181 | 0.0342660 | 0.3865248 | 0.0629512 |
| | | | 0.00005 | 3107 | 0.0344384 | 0.3794326 | 0.0631455 |
| | | | 0.5 | 3206 | 0.0339988 | 0.3865248 | 0.0625000 |
| | | | 50 | 3539 | 0.0302345 | 0.3794326 | 0.0560063 |
| 23 | batch | recursive | 0.0000005 | 3148 | 0.0320839 | 0.3581560 | 0.0588921 |
| | | | 0.00005 | 3159 | 0.0319721 | 0.3581560 | 0.0587039 |
| | | | 0.5 | 3284 | 0.0310597 | 0.3617021 | 0.0572070 |
| | | | 50 | 3617 | 0.0270943 | 0.3475177 | 0.0502693 |
| 24 | batch | recursive | 0.0000005 | 3134 | 0.0331844 | 0.3687943 | 0.0608899 |
| | | | 0.00005 | 3061 | 0.0325537 | 0.3652482 | 0.0597795 |
| | | | 0.5 | 3234 | 0.0312307 | 0.3581560 | 0.0574516 |
| | | | 50 | 3489 | 0.0286615 | 0.3546099 | 0.0530363 |
| 25 | batch | recursive | 0.0000005 | 3073 | 0.0357956 | 0.3900709 | 0.0655738 |
| | | | 0.00005 | 3061 | 0.0329958 | 0.3581560 | 0.0604248 |
| | | | 0.5 | 3232 | 0.0315594 | 0.3617021 | 0.0580535 |
| | | | 50 | 3518 | 0.0275725 | 0.3439716 | 0.0510526 |
| 26 | batch | recursive | 0.0000005 | 3120 | 0.0323718 | 0.3581560 | 0.0593768 |
| | | | 0.00005 | 3110 | 0.0331190 | 0.3652482 | 0.0607311 |
| | | | 0.5 | 3221 | 0.0313567 | 0.3581560 | 0.0576649 |
| | | | 50 | 3563 | 0.0291889 | 0.3687943 | 0.0540962 |
| 27 | batch | recursive | 0.0000005 | 3162 | 0.0328906 | 0.3687943 | 0.0603949 |
| | | | 0.00005 | 3211 | 0.0330115 | 0.3758865 | 0.0606928 |
| | | | 0.5 | 3307 | 0.0317508 | 0.3723404 | 0.0585121 |
| | | | 50 | 3640 | 0.0277473 | 0.3581560 | 0.0515043 |
| 28 | batch | recursive | 0.0000005 | 3174 | 0.0330813 | 0.3723404 | 0.0607639 |
| | | | 0.00005 | 3178 | 0.0314663 | 0.3546099 | 0.0578035 |
| | | | 0.5 | 3297 | 0.0303306 | 0.3546099 | 0.0558815 |
| | | | 50 | 3544 | 0.0299097 | 0.3758865 | 0.0554104 |
| 29 | batch | recursive | 0.0000005 | 3066 | 0.0352250 | 0.3829787 | 0.0645161 |
| | | | 0.00005 | 3139 | 0.0328130 | 0.3652482 | 0.0602163 |
| | | | 0.5 | 3232 | 0.0324876 | 0.3723404 | 0.0597610 |
| | | | 50 | 3585 | 0.0281729 | 0.3581560 | 0.0522369 |
| 30 | batch | recursive | 0.0000005 | 3182 | 0.0317410 | 0.3581560 | 0.0583141 |
| | | | 0.00005 | 3154 | 0.0317058 | 0.3546099 | 0.0582072 |
| | | | 0.5 | 3272 | 0.0311736 | 0.3617021 | 0.0574001 |
| | | | 50 | 3585 | 0.0292887 | 0.3723404 | 0.0543057 |

## Anexo D. Resultados de experimentos para selección de valores de los parámetros de Linguistica

| Parámetros | | | | Resultados | | |
|---|---|---|---|---|---|---|
| ***max_word_types*** | ***min_stem_length*** | ***max_affix_length*** | ***min_sig_count*** | **Total de morfemas identificados** | ***precision*** | ***recall*** |
| por defecto | por defecto | por defecto | por defecto | Media: 1799.73<br>Desv.Estándar: 29.53424<br>Err.Estándar: 0.25536 | Media: 0.02232<br>Desv.Estándar: 0.00158<br>Err.Estándar: 0.00001 | Media: 0.14243<br>Desv.Estándar: 0.00968<br>Err.Estándar: 0.00008 |
| 13 377 | por defecto | por defecto | por defecto | Media: 1799.73<br>Desv.Estándar: 29.53424<br>Err.Estándar: 0.25536 | Media: 0.02232<br>Desv.Estándar: 0.00158<br>Err.Estándar: 0.00001 | Media: 0.14243<br>Desv.Estándar: 0.00968<br>Err.Estándar: 0.00008 |
| por defecto | 1 | por defecto | por defecto | Media: 2027.87<br>Desv.Estándar: 38.10850<br>Err.Estándar: 0.32949 | Media: 0.02839<br>Desv.Estándar: 0.00133<br>Err.Estándar: 0.00001 | Media: 0.20414<br>Desv.Estándar: 0.00975<br>Err.Estándar: 0.00008 |
| por defecto | por defecto | 15 | por defecto | Media: 2579.93<br>Desv.Estándar: 35.91747<br>Err.Estándar: 0.31055 | Media: 0.01618<br>Desv.Estándar: 0.00121<br>Err.Estándar: 0.00001 | Media: 0.14799<br>Desv.Estándar: 0.01094<br>Err.Estándar: 0.00009 |
| por defecto | por defecto | 28 | por defecto | Media: 2579.93<br>Desv.Estándar: 35.91747<br>Err.Estándar: 0.31055 | Media: 0.01618<br>Desv.Estándar: 0.00121<br>Err.Estándar: 0.00001 | Media: 0.14799<br>Desv.Estándar: 0.01094<br>Err.Estándar: 0.00009 |
| por defecto | por defecto | por defecto | 2 | Media: 2246.43<br>Desv.Estándar: 30.95122<br>Err.Estándar: 0.26761 | Media: 0.02680<br>Desv.Estándar: 0.00104<br>Err.Estándar: 0.00001 | Media: 0.21348<br>Desv.Estándar: 0.00814<br>Err.Estándar: 0.00007 |

| | | | | | | |
|---|---|---|---|---|---|---|
| por defecto | por defecto | por defecto | 3 | Media: 2038.80<br>Desv.Estándar: 35.91983<br>Err.Estándar: 0.31057 | Media: 0.02461<br>Desv.Estándar: 0.00135<br>Err.Estándar: 0.00001 | Media: 0.17790<br>Desv.Estándar: 0.00950<br>Err.Estándar: 0.00008 |
| por defecto | por defecto | por defecto | 4 | Media: 1902.73<br>Desv.Estándar: 34.33299<br>Err.Estándar: 0.29685 | Media: 0.02318<br>Desv.Estándar: 0.00128<br>Err.Estándar: 0.00001 | Media: 0.15638<br>Desv.Estándar: 0.00835<br>Err.Estándar: 0.00007 |
| por defecto | por defecto | por defecto | 6 | Media: 1716.17<br>Desv.Estándar: 30.86688<br>Err.Estándar: 0.26688 | Media: 0.02173<br>Desv.Estándar: 0.00166<br>Err.Estándar: 0.00001 | Media: 0.13215<br>Desv.Estándar: 0.00949<br>Err.Estándar: 0.00008 |
| 1 000 | 1 | 4 | 2 | Media: 2551.86<br>Desv.Estándar: 32.16738<br>Err.Estándar: 0.27812 | Media: 0.02817<br>Desv.Estándar: 0.00092<br>Err.Estándar: 0.00001 | Media: 0.25485<br>Desv.Estándar: 0.00755<br>Err.Estándar: 0.00007 |

## Anexo E. Resultados de experimentos para selección de valores del parámetro *merge_operations* para BPE

| **Parámetros** | **Resultados** | | |
|---|---|---|---|
| ***merge operations*** | **Total de morfemas identificados** | ***precision*** | ***recall*** |
| 100 | Media: 123.87<br>Desv.Estándar: 1.65536<br>Err.Estándar: 0.01431 | Media: 0.33185<br>Desv.Estándar: 0.00924<br>Err.Estándar: 0.00008 | Media: 0.14574<br>Desv.Estándar: 0.00388<br>Err.Estándar: 0.00003 |
| 200 | Media: 201.40<br>Desv.Estándar: 2.28337<br>Err.Estándar: 0.01974 | Media: 0.30143<br>Desv.Estándar: 0.00739<br>Err.Estándar: 0.00006 | Media: 0.21525<br>Desv.Estándar: 0.00476<br>Err.Estándar: 0.00004 |
| 300 | Media: 286.17<br>Desv.Estándar: 2.60084<br>Err.Estándar: 0.02249 | Media: 0.23823<br>Desv.Estándar: 0.00636<br>Err.Estándar: 0.00006 | Media: 0.24173<br>Desv.Estándar: 0.00582<br>Err.Estándar: 0.00005 |
| 400 | Media: 374.47<br>Desv.Estándar: 3.58862<br>Err.Estándar: 0.03103 | Media: 0.20547<br>Desv.Estándar: 0.00530<br>Err.Estándar: 0.00005 | Media: 0.27281<br>Desv.Estándar: 0.00651<br>Err.Estándar: 0.00006 |
| 500 | Media: 464.00<br>Desv.Estándar: 3.22704<br>Err.Estándar: 0.02790 | Media: 0.17960<br>Desv.Estándar: 0.00428<br>Err.Estándar: 0.00004 | Media: 0.29551<br>Desv.Estándar: 0.00686<br>Err.Estándar: 0.00006 |
| 600 | Media: 553.93<br>Desv.Estándar: 3.58092<br>Err.Estándar: 0.03096 | Media: 0.15984<br>Desv.Estándar: 0.00462<br>Err.Estándar: 0.00004 | Media: 0.31395<br>Desv.Estándar: 0.00842<br>Err.Estándar: 0.00007 |
| 700 | Media: 644.40<br>Desv.Estándar: 5.13675<br>Err.Estándar: 0.04441 | Media: 0.14422<br>Desv.Estándar: 0.00435<br>Err.Estándar: 0.00004 | Media: 0.32955<br>Desv.Estándar: 0.00998<br>Err.Estándar: 0.00009 |
| 800 | Media: 733.67<br>Desv.Estándar: 5.38410<br>Err.Estándar: 0.04655 | Media: 0.13158<br>Desv.Estándar: 0.00398<br>Err.Estándar: 0.00003 | Media: 0.34232<br>Desv.Estándar: 0.00993<br>Err.Estándar: 0.00009 |
| 900 | Media: 822.87<br>Desv.Estándar: 6.02714<br>Err.Estándar: 0.05211 | Media: 0.12189<br>Desv.Estándar: 0.00334<br>Err.Estándar: 0.00003 | Media: 0.35567<br>Desv.Estándar: 0.00977<br>Err.Estándar: 0.00008 |
| 1 000 | Media: 911.87<br>Desv.Estándar: 6.77080<br>Err.Estándar: 0.05854 | Media: 0.11339<br>Desv.Estándar: 0.00265<br>Err.Estándar: 0.00002 | Media: 0.36667<br>Desv.Estándar: 0.00901<br>Err.Estándar: 0.00008 |

## Anexo F. Representaciones vectoriales en enfoque multilingüe

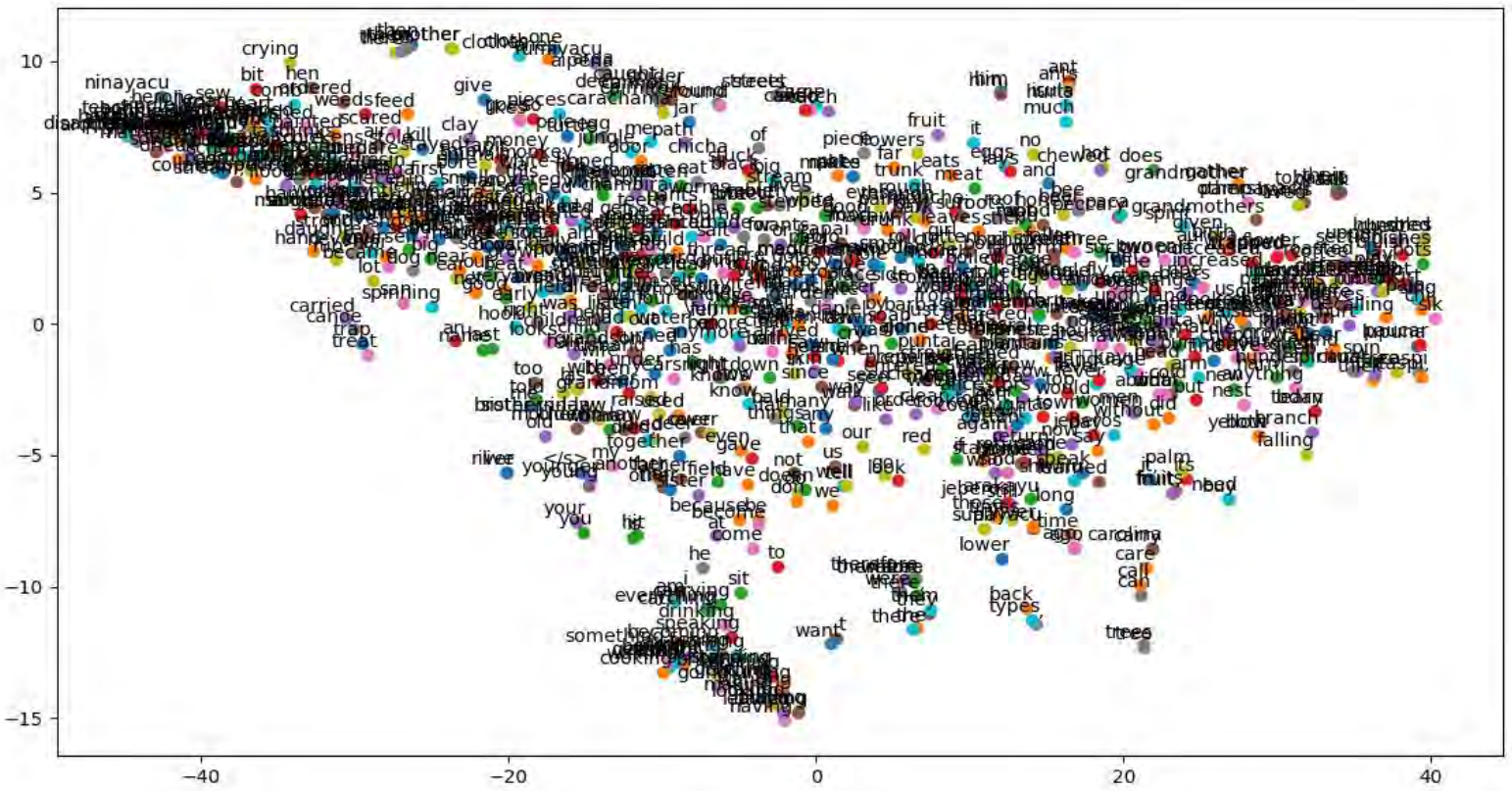


**Figura 16** Espacio vectorial de palabras en inglés

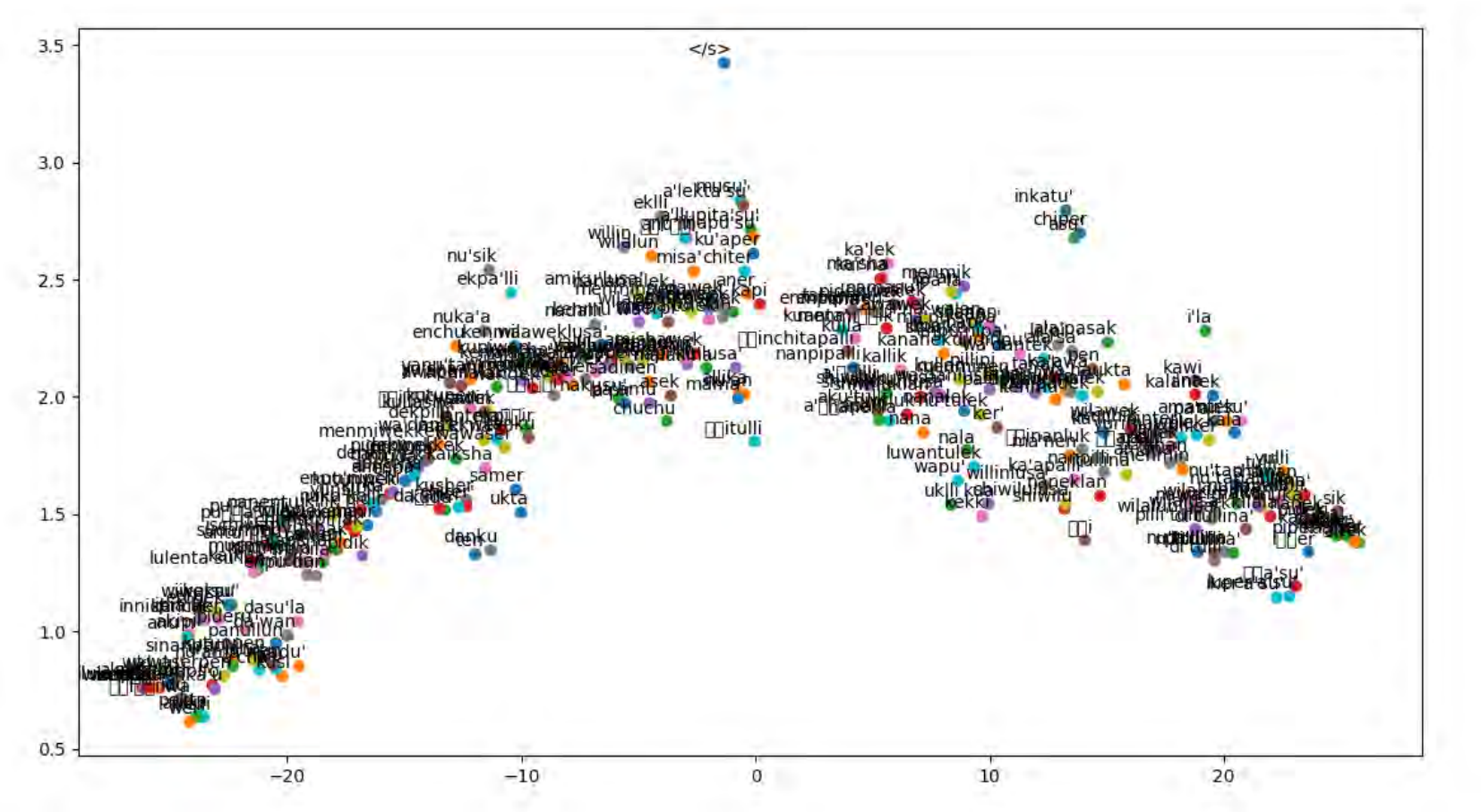


**Figura 17** Espacio vectorial de palabras en shiwilu representado en 2 dimensiones, luego de ser alineado con el espacio vectorial de las palabras en inglés

ET LUX IN TENEBRIS LUCET
MCMXVII