UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Construcción de Recursos para Traducción Automática Guaraní-Español

Proyecto de Grado

Nicolás Giossa - Santiago Góngora

Tutor: Luis Chiruzzo

Carrera de Ingeniería en Computación
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Setiembre de 2021

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE INFORME

Aiala Rosá

Jorge Visca

Mathías Etcheverry

Montevideo – Uruguay
Setiembre de 2021

A todas las personas que mantienen viva la
cultura de los pueblos originarios.

Agradecimientos

En primer lugar queremos agradecer a nuestras familias (en sentido amplio) por apoyarnos durante todos estos años de camino, así como a *las Agustinas* por la ayuda y la paciencia.

A nuestros amigos Maxi, Renzo, Masi, Jojo y Lucas por compartir muchos años de largas jornadas de estudio y, cómo no, algunas horas de esparcimiento. A otros grandes compañeros de estudio y trabajo que tuvimos, con los que aprendimos grandes cosas, y a otros amigos de la vida.

Nuestro agradecimiento a Marvin Agüero-Torales por ayudarnos a traducir uno de los *tests* utilizados, luego de contactarnos en la `NAACL2021` al ver nuestro trabajo, y al grupo de PLN de la facultad por todos los consejos brindados.

Y especialmente a Luis, por guiarnos durante todo el proyecto, respetando nuestros tiempos y permitiéndonos experimentar sobre ciertas aristas que no queríamos dejar de explorar.

Más allá de esta lista breve y general, cada uno de nuestros allegados sabe el granito de arena que aportó para que lleguemos hasta acá, pues los pequeños gestos generan grandes cambios.

¡Aguyje!

"Dice que él también es un rey y que
tampoco escuchará."
Gabriel, traduciendo la voluntad
de los guaraníes.

The mission (1986)

RESUMEN

Aunque han pasado varios siglos desde que el guaraní y el español entraron en contacto por primera vez, la traducción automática entre este par de lenguas se mantiene como un tópico de investigación poco explorado dentro de la comunidad científica. Incluso si no nos limitamos a la traducción automática y consideramos todas las tareas de PLN, la escasez de recursos desarrollados para el guaraní se evidencia más aún. Por estar en esta situación, el guaraní es considerada una *lengua de escasos recursos*.

Este trabajo presenta una serie de esfuerzos realizados para fortalecer el desempeño de la traducción automática entre el guaraní y el español, tanto en una dirección como en la otra.

Al inicio de nuestra investigación construimos un conjunto paralelo de noticias y uno monolingüe conformado por *tweets*. El conjunto paralelo construido contiene 15.175 pares de oraciones, mientras que el conjunto monolingüe cuenta con 9.635 *tweets*.

Posteriormente nos centramos en construir representaciones vectoriales de palabras y de su uso, junto al resto de recursos construidos, a la hora de realizar experimentos de traducción automática. En cuanto a los resultados experimentales, si bien los vectores de palabras logran tener buenos resultados tanto en *tests* intrínsecos como extrínsecos, creemos que la ausencia de diversidad en los textos impacta muy fuertemente en su calidad. Respecto a la traducción automática logramos mejorar resultados previos en la dirección guaraní-español; adicionalmente comparamos nuestro desempeño en el sentido español-guaraní con el de los participantes de la *shared task* de AmericasNLP, que tuvo lugar en junio de 2021.

Palabras claves: Guaraní, PLN, Procesamiento de Lenguaje Natural, Traducción automática, Vectores de palabras, Word embeddings, OpenNMT.

Tabla de contenidos

1 Introducción **1**

1.1 El Guaraní y su situación en la región 1

1.2 Motivacion de nuestra investigación 4

1.3 Objetivos 5

1.4 Estructura del documento 6

2 Marco Teórico **7**

2.1 Características del guaraní 7

2.2 Procesamiento de Lenguaje Natural 9

2.2.1 Aprendizaje automático aplicado al PLN 12

2.3 Traducción Automática 14

2.3.1 Enfoques basados en reglas 15

2.3.2 Enfoques estadísticos 15

2.3.3 Enfoques basados en redes neuronales 16

2.3.4 Evaluación automática de traducciones 17

2.4 Vectores de palabras 20

2.4.1 Evaluación 21

2.4.2 Test de Similitud 23

2.5 PLN para el Guaraní 24

3 Construcción del Corpus **26**

3.1 Conjunto paralelo 26

3.1.1 Crawling general en sitios de Paraguay 27

3.1.2 Crawling particular en sitios de noticias 27

3.1.3 Conjunto final 29

3.2 Conjunto de tweets 30

3.2.1 Obtención de *tweets* a partir de palabras clave 31

3.2.2 Reconocimiento de guaraní 32

3.2.3 Conjunto final . . . 37
3.3 Otros textos . . . 40

4 Vectores de Palabras **41**
4.1 Entrenamiento . . . 41
4.2 Tests de analogías y similitud . . . 43
4.2.1 Test de analogías: Family . . . 43
4.2.2 Test de analogías: Capital-Common-Countries . . . 44
4.2.3 Test de similitud: MC-30 . . . 45
4.2.4 Resultados . . . 45
4.3 Experimentos de visualización de palabras . . . 49
4.3.1 Resultados . . . 49

5 Traducción automática **54**
5.1 Construcción de modelos . . . 54
5.2 Guaraní-español . . . 56
5.2.1 Resultados . . . 58
5.2.2 Análisis de ejemplos de traducción . . . 60
5.3 Español-guaraní . . . 62
5.3.1 Resultados . . . 63
5.4 Back-Translation . . . 64
5.5 Relación entre la calidad de la traducción y el largo de la oración 66

6 Conclusiones **69**
6.1 Conclusiones del proyecto . . . 69
6.2 Recursos disponibles . . . 70
6.3 Trabajo futuro . . . 72
6.3.1 Sobre el corpus . . . 72
6.3.2 Sobre los vectores de palabras y los tests intrínsecos . . . 73
6.3.3 Sobre la traducción automática . . . 74

Bibliografía **76**

Glosario **86**

Capítulo 1

Introducción

El Guaraní es una de las lenguas indígenas que aún sobreviven en Latinoamérica, teniendo la particularidad de ser hablada por más de diez millones de personas y de haber sido reconocida como una de las lenguas oficiales del Mercosur[1].

En el contexto del trabajo final para la obtención del título de Ingeniería en Computación, presentamos aquí nuestros aportes al desarrollo de herramientas computacionales con el objetivo principal de fortalecer la *performance* de los modelos de traducción automática Guaraní-Español.

Parte de este trabajo fue publicado y presentado en el primer *workshop* de PLN para lenguas indígenas de las Américas (Góngora et al. 2021), organizado junto a la NAACL2021[2].

1.1. El Guaraní y su situación en la región

Con más de 10 millones de hablantes en Argentina, Bolivia, Brasil y principalmente Paraguay, el guaraní es una de las lenguas indígenas habladas en América del Sur y tiene la particularidad de ser una de las pocas lenguas indígenas utilizadas diariamente en la comunicación de la población. En la

[1] https://www.parlamentomercosur.org/innovaportal/v/8221/2/parlasur/lengua-guarani-se-convierte-en-idioma-oficial-de-trabajo-del-parlamento-del-mercosur.html - Accedido por última vez el 22.08.2021.

[2] https://aclanthology.org/volumes/2021.americasnlp-1/ - Accedido por última vez el 22.08.2021.

figura 1.1 podemos ver un póster elaborado por el Instituto nacional de estadística de Paraguay[1] con el objetivo de visibilizar la importancia del guaraní para la población, según los datos del censo continuo de 2019. Como podemos observar, casi el 70 % de la población habla guaraní dentro de su hogar y un 38,5 % lo usa exclusivamente.

Figura 1.1: Póster informativo del Instituto nacional de estadística de Paraguay.

Si bien la lengua ya era hablada por habitantes nativos de la región, recién durante las misiones jesuíticas se realizaron las primeras formalizaciones de ella. Un ejemplo de esto es la publicación en 1696 de *Arte de la lengua guaraní* [2], escrita por el sacerdote Pablo Restivo. En la figura 1.2 se puede ver la carátula de la edición de 1724.

Siglos después de las misiones jesuíticas, el guaraní se mantiene como una

[1] https://www.ine.gov.py/news/news-contenido.php?cod-news=507 - Accedido por última vez el 22.08.2021.

[2] http://www.cervantesvirtual.com - Accedido por última vez el 22.08.2021.

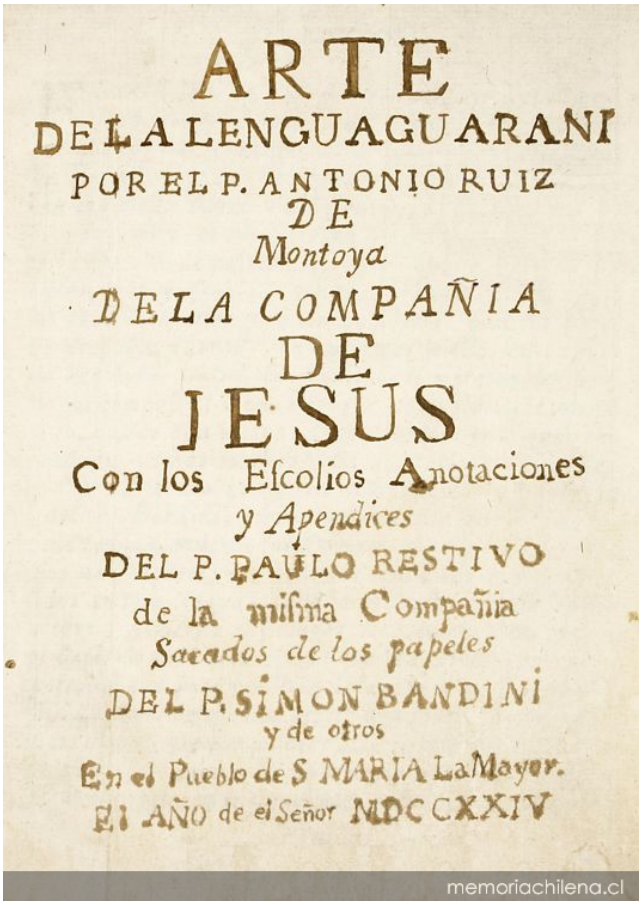


Figura 1.2: Carátula de *Arte de la lengua guaraní* (1724).

lengua viva, lo cual se evidencia en la cantidad de hablantes en Paraguay, tanto monolingües como bilingües. Sin embargo, culturalmente, hablar una lengua indígena no fue siempre bien recibido[1] (Vitale, 2012). Para cambiar esta realidad se han promovido políticas como la declaración del guaraní como *Lengua Oficial del Paraguay* en la Constitución de 1992[2] y su fortalecimiento en la enseñanza[3] o la inclusión del Guaraní como una de las lenguas oficiales del Mercosur[4].

[1] https://www.nytimes.com/es/2018/01/08/espanol/america-latina/paraguay-guarani-espanol-lengua-indigena.html - Accedido por última vez el 22.08.2021.

[2] https://www.oas.org/juridico/spanish/par_res3.htm - Accedido por última vez el 22.08.2021.

[3] https://www.bacn.gov.py/leyes-paraguayas/2895/de-lenguas - Accedido por última vez el 22.08.2021.

[4] https://www.parlamentomercosur.org/innovaportal/v/8221/2/parlasur/lengua-guarani-se-convierte-en-idioma-oficial-de-trabajo-del-parlamento-del-mercosur.html - Accedido por última vez el 22.08.2021.

1.2. Motivacion de nuestra investigación

Históricamente la investigación en Procesamiento de Lenguaje Natural (PLN) y Lingüística Computacional se ha centrado principalmente en el estudio del idioma inglés. Si bien hay, en menor medida, diversidad de enfoques y recursos disponibles para una gran cantidad de otras lenguas, la situación para las lenguas minoritarias es considerablemente peor.

Joshi et al. (2020) realizaron una taxonomía[1] que clasifica a los lenguajes en función de su presencia en conferencias de PLN, así como de los recursos disponibles. Los resultados obtenidos confirman la percepción de la comunidad: tanto el inglés como el español se encuentran en la cima de la taxonomía, mientras que lenguas indígenas como el náhuatl, el quechua y el aymara se ubican en la penúltima categoría. Tanto el tzeltal, el mixtecto y el tsotsil como el guaraní se encuentran en la última categoría de la taxonomía. A estas lenguas con escasos o nulos recursos desarrollados se las suele denominar *low-resource languages* (lenguas de escasos recursos).

La investigación lingüística y computacional sobre una lengua no solamente aporta el valor aislado de sus resultados, sino que además provee a la comunidad hablante —y no hablante— de herramientas para seguir manteniendo estas lenguas en el mapa. Está claro que, más allá de la soberanía cultural que las herramientas desarrolladas puedan aportar, lo principal es seguir promoviendo políticas culturales y educativas sobre estas lenguas para que se mantengan vivas. Sin ir más lejos, por diversas razones, casi no quedaron registros lingüísticos[2] de las lenguas habladas por los indígenas que habitaban la región del Río de la plata (Rosa, 2013), por lo que trabajar sobre ellas es entonces muy difícil.

Sin embargo este no es el caso del guaraní que, como mencionamos, está reconocida como una de las lenguas oficiales del Mercosur y tiene un papel protagónico en la comunicación diaria del Paraguay, lo que habilita la recolección de textos generados en el día a día. Siguiendo esta línea intentaremos,

[1] https://microsoft.github.io/linguisticdiversity/assets/lang2tax.txt - Accedido por última vez el 22.08.2021.

[2] Solamente se cuenta con algunas colecciones léxicas y frases aisladas, que no permiten la generalización de una gramática completa.

desde la computación, fortalecer con nuestro trabajo los incansables esfuerzos de mantener al guaraní entre las lenguas vivas del continente.

1.3. Objetivos

El objetivo general del proyecto es, entonces, continuar con los esfuerzos de construcción de recursos que potencien los resultados de la traducción automática guaraní-español. Puntualmente, los objetivos del proyecto son:

1. Recolectar la mayor cantidad posible de texto con contenido en guaraní. En lo posible, que el texto tenga su parte correspondiente en español, de modo de ampliar la cantidad de texto paralelo para modelos de traducción automática. Sin embargo, el texto monolingüe es también bienvenido y de utilidad.

2. Realizar experimentos con vectores de palabras que permitan acercarnos a una medición de la calidad de los corpus recopilados.

3. Realizar experimentos de traducción automática buscando mejorar la *performance* de esta tarea.

1.4. Estructura del documento

En esta sección detallamos la estructura del documento y cómo se compone cada uno de sus capítulos.

En el capítulo 2 presentaremos los conceptos teóricos necesarios para leer el informe. Muchos métodos y tests allí expuestos se dan por conocidos a lo largo del documento.

En el capítulo 3 describiremos la primera etapa de nuestro trabajo, que fue la expansión del corpus paralelo con el que contábamos inicialmente, presentado en Chiruzzo et al. 2020. Detallaremos el proceso de construcción de un conjunto paralelo conformado por noticias y uno monolingüe compuesto por texto de la red social Twitter. Detallaremos también algunos recursos monolingües extraídos de otras fuentes.

En los capítulos 4 y 5 presentaremos diversos experimentos realizados sobre este texto. En primer lugar los que usan texto monolingüe, la construcción de vectores de palabras y su evaluación. Luego presentaremos los resultados de los experimentos de traducción automática en ambos sentidos, guaraní-español y español-guaraní.

Finalmente en el capítulo 6 presentaremos las conclusiones de nuestro trabajo. Enumeraremos cada uno de los recursos generados y también desarrollaremos algunas ideas de trabajo futuro.

Capítulo 2

Marco Teórico

En este capítulo describiremos el marco teórico que guió la investigación, con el objetivo de que el lector pueda comprender alguna de las decisiones tomadas durante el desarrollo del proyecto. Haremos una breve presentación del guaraní para luego introducir el procesamiento de lenguaje natural, la traducción automática, los vectores de palabras y su evaluación. Por último desarrollaremos el trabajo relevante a la fecha en PLN para el guaraní.

2.1. Características del guaraní

Como mencionamos en la introducción, el guaraní es una lengua de origen autóctono hablada actualmente en ciertas regiones de Sudamérica por más de diez millones de personas. Forma parte de la familia *tupí-guaraní*, cuyo surgimiento se remonta entre 3.000 y 4.000 años atrás (Michael et al. 2015) en los territorios que hoy son Argentina, Brasil, Bolivia, Perú y Paraguay (como se ve en la figura 2.1).

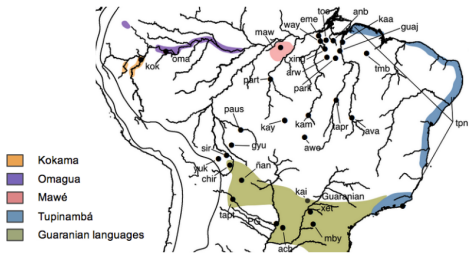


Figura 2.1: Mapa de la distribución geográfica de las lenguas pertenecientes a la familia tupí-guaraní, extraída de Michael et al. 2015.

En este trabajo nos centraremos en el guaraní hablado en Paraguay que, como se mostró en la figura 1.1, es una parte esencial de la comunicación en ese país. Como desarrolla Lustig (2010), el guaraní hablado en Paraguay no se corresponde exactamente al original pues ha experimentado más de cuatro siglos de contacto con el español. Si bien este *guaraní puro* ha sido rescatado en gran parte gracias a los escritos generados durante las misiones jesuíticas (siglos XVII-XVIII), en Paraguay está muy extendido el dialecto denominado jopará. Este dialecto es la evidencia explícita del proceso de contacto que describe Lustig, y se caracteriza por su nivel dispar de mezcla con el español. Chiruzzo et al. (2020) presentan cuatro grados de mezcla en el jopará de la siguiente manera[1]:

- Expresiones "**puras**" en guaraní, sin rastros de español. Por ejemplo: "yvágaicha hovy" (azul como el cielo)
- Expresiones en guaraní con **neologismos** del español pero adaptados a su morfología. Por ejemplo: "¡che mo renegaite la i porte! " (me hace renegar su actitud), donde el morfema "renega" es una palabra del español y el sufijo "ite" aumenta su intensidad (Academia de la Lengua Guaraní, 2018).
- Expresiones en guaraní con palabras **prestadas** del español. Por ejemplo: "Che ru músico" (Mi padre es músico).
- Expresiones en español con palabras **prestadas** del guaraní. Por ejemplo: "no tengo ite gana de hablá " (no tengo absolutamente ganas de hablar), donde "ite" es el morfema que actúa como sufijo para indicar el grado superlativo, pero en este caso actúa como una palabra aislada.

En el segundo ejemplo vimos que se usa "renegaite" para decir "renegar mucho", y que es el producto de tomar la raíz *renega* y concatenarle el sufijo *ite* que indica el grado superlativo. Por esta característica, de unir morfemas para crear palabras y expresiones, se considera que el guaraní es una lengua polisintética y aglutinante (Estigarribia y Pinta, 2017).

Como podemos ver, las características del guaraní lo hacen muy diferente al español. Por poner otro ejemplo, como lo indica la "Gramática oficial del guaraní" (Academia de la Lengua Guaraní, 2018), el alfabeto del guaraní

[1]Los ejemplos son tomados textualmente del trabajo de Lustig (2010).

contiene 33 letras, muchas de ellas no presentes en el del español. Un ejemplo claro es la consonante **puso**, que se representa con la comilla simple (') y la misma gramática define su sonido como "una pausa breve, producida por la obstrucción del flujo de aire en la glotis, antes de la pronunciación de la vocal con la que forma sílaba.".

Desde el punto de vista de la computación, estas características lingüísticas hacen particularmente difícil su procesamiento automático, pues existen múltiples combinaciones de morfemas cuyo significado puede variar según el contexto. Esto también genera que haya muchísimas palabras, razón por la cual los diccionarios de guaraní solo suelen tener las palabras básicas (Zarratea, 2009). Esta característica hace que los corpus presenten una cantidad muy baja de ocurrencias por palabra, por lo que los enfoques estadísticos se ven afectados, más que en otras lenguas, debido a la falta de cantidad de texto y diversidad léxica.

Aunque no tanto como el guaraní, el español también tiene una morfología muy rica, lo que genera que la traducción automática entre este par de lenguas (guaraní y español) sea especialmente difícil de resolver. Una posible forma de abordar este problema es mediante la explotación de características morfológicas, como es el caso del proyecto de fin de carrera de Borges y Mercant (2019), que se describe al final de este capítulo.

2.2. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (PLN) (o *Lingüística computacional*)[1] es una subárea de la Inteligencia Artificial y de las Ciencias de la Computación que se ocupa del desarrollo y evaluación de métodos computacionales que permitan **comprender**, **analizar** y **generar** lenguaje natural. Por *lenguaje natural* entendemos cualquier lenguaje que sea hablado por humanos, en contraposición con los lenguajes formales que son creados para un determinado fin y generalmente no presentan ambigüedad, como por ejemplo las direcciones de correo electrónico o los lenguajes de programación. De hecho, una de las principales dificultades en la disciplina de PLN es lidiar con la ambigüedad,

[1] https://www.aclweb.org/portal/what-is-cl - Accedido por última vez el 22.08.2021.

que está presente en todos los lenguajes naturales.

Existe gran diversidad de problemas que el PLN intenta resolver. Una forma de clasificar estos objetos de estudio es identificando diferentes tareas (*tasks* en inglés) que se intentan resolver como parte del procesamiento del lenguaje natural. En primer lugar tenemos tareas que son más interiores al análisis de la lengua y suelen servir como herramientas para otras tareas. Entre ellas podemos destacar:

- **Análisis morfológico**: Los morfemas son las unidades más pequeñas que pueden tener sentido en una lengua. Esta tarea busca analizar su uso y su combinación para formar las palabras (Minnen et al. 2001).

- **Part-of-speech tagging** (POS): Esta tarea se ocupa de la siguiente unidad de estudio, la palabra. El *POS tagging* busca asignarle a cada palabra de la entrada la categoría léxica que le corresponde, como verbo, adverbio, nombre, adjetivo, etc. (Jurafsky y Martin, 2021). Por ejemplo, para la oración "Ella ama el chocolate" las etiquetas esperadas son: (*Ella*, nombre), (*ama*, verbo), (*el*, determinante), (*chocolate*, nombre).

- **Análisis sintáctico**: El siguiente nivel de análisis se da en agrupar las palabras en unidades mayores como las frases u oraciones. Esta tarea busca analizar una entrada según un formalismo sintáctico, como por ejemplo el análisis de constituyentes (Jurafsky y Martin, 2021).

- **Named-entity recognition** (NER): En este caso se intenta encontrar cuáles palabras o expresiones constituyen una *entidad con nombre*. Entendemos por *entidad con nombre* a todo concepto del mundo real que pueda ser identificado por un nombre propio (Jurafsky y Martin, 2021). Por ejemplo, en la oración "Diego Forlán nació en la República Oriental del Uruguay." se pretende identificar que las entidades con nombre son "Diego Forlán" y "República Oriental del Uruguay", donde la primera corresponde a una persona y la segunda a un lugar.

- **Análisis semántico**: Este análisis trata, entre otras cosas, de encontrar el significado de las palabras y cómo se pueden componer para construir el significado de las oraciones mediante *representaciones del significado*. Un

enfoque usado ha sido, por ejemplo, la lógica de primer orden (Jurafsky y Martin, 2009).

- **Análisis de sentimientos**: El objetivo es determinar si un texto presenta una opinión positiva o negativa (Jurafsky y Martin, 2009). Una pequeña variación de este problema es el del **análisis de emociones**, donde lo que se busca es determinar por ejemplo si el escritor expresa enojo, felicidad, empatía, etc. La competencia TASS (García-Vega et al. 2020) es un ejemplo de evento recurrente sobre análisis de sentimientos para *tweets* en español.

- **Detección de discurso de odio**: Esta tarea es similar a la anterior, pero se enfoca en lenguaje que expresa odio hacia determinados grupos sociales. Un ejemplo de este trabajo es el proyecto de grado de Kunc y Saravia (2020).

Por último existen tareas de más alto nivel que son generalmente útiles a usuarios finales. Estas tareas suelen integrar varias herramientas que resuelven las tareas previamente descritas. Son muchísimos los ejemplos de estas aplicaciones pero, por mencionar solo algunos, tenemos:

- **Sistemas conversacionales**: Son sistemas que mantienen un diálogo fluido con el usuario, emulando el comportamiento de la conversación humana, ya sea en dominios abiertos o cerrados (Jurafsky y Martin, 2021).

- **Sistemas de creatividad computacional**: Estos sistemas buscan generar salidas o construir agentes que se comporten de manera creativa ante la observación humana (Colton y Wiggins, 2012). Algunos objetivos de estos sistemas son, por ejemplo, la generación de narrativa (Alabdulkarim et al. 2021; Gervás et al. 2019) y poesía (Van de Cruys, 2020), o los agentes interactivos para videojuegos, como AIDungeon[1].

Si bien estas son algunas de las tantas tareas abordadas en el área, quizás el mejor ejemplo que podemos dar es el que intentamos resolver en este trabajo. Esta tarea es la **traducción automática** y la desarrollaremos en la sección 2.3, pero antes introduciremos algunos conceptos que son relevantes al trabajar hoy en día en procesamiento de lenguaje natural.

[1] https://play.aidungeon.io/main/landing - Accedido por última vez el 22.08.2021.

2.2.1. Aprendizaje automático aplicado al PLN

En el pasado, trabajar con heurísticas y reglas diseñadas manualmente era la práctica estándar en el PLN. Sin embargo, debido a sus complejidades y limitaciones, ya hace varias décadas que esta situación comenzó a cambiar drásticamente. Desde los años 2000 lo más usado para resolver la gran mayoría de tareas son los métodos de *Machine Learning* (aprendizaje automático) (Jurafsky y Martin, 2009).

El área de *Machine Learning* estudia aquellos algoritmos que aprenden mediante la experiencia. Mitchell (1997) define que un programa "aprende de la experiencia E con respecto a cierta clase de tareas T y medida P, si su *performance* en la tarea T — evaluada según la medida P — mejora con la experiencia E."

Para ejemplificar esta definición, apliquémosla sobre la tarea de *análisis de sentimientos* definida previamente, que consiste en determinar si un texto presenta polaridad positiva o negativa. Decimos que nuestro algoritmo **aprende**, al mirar ejemplos de textos y su polaridad, a identificar la polaridad de un texto cualquiera, si al medirlo con ciertas **medidas** su capacidad de identificar la polaridad mejora. Notemos que para que el algoritmo logre aprender es necesario que tenga **ejemplos** (*datos*).

En el PLN, a estas colecciones de ejemplos (en forma de texto) las llamamos **corpus** y, por lo general, tienen características que los identifican (Jurafsky y Martin, 2021). Por ejemplo, Gebru et al. (2018) proponen siempre detallar algunas como la motivación de creación, la metodología de construcción o la variedad lingüística presente. A su vez, un corpus puede contener información extra tales como las categorías léxicas presentes, o las entidades con nombres mencionadas en él, en cuyo caso decimos que se trata de un **corpus anotado**. Continuando con el análisis de sentimientos, es común que los textos usados para entrenar esos algoritmos estén anotados con la polaridad correspondiente, como por ejemplo:

- "¡Cómo me encantan estas galletitas!" $\longrightarrow$ Positivo
- "¡¿Cómo puede ser que te gusten esas galletitas?!" $\longrightarrow$ Negativo

A este problema, de clasificar a una observación entre un conjunto de categorías discretas, se le suele llamar *problema de clasificación* (Jurafsky y Martin, 2009) y existen muchos modelos para resolverlo, como *Naive Bayes*, *Support-vector Machines* o *Redes neuronales*.

Otro componente presente en la definición de Mitchell (1997) es la evaluación de *performance* del algoritmo según una **medida**. Por ejemplo, para el problema de clasificación en análisis de sentimientos, las medidas clásicas son (Jurafsky y Martin, 2021):

- *Precision* (precisión), que mide cuántas observaciones identificadas como *positivas* lo eran realmente, de entre el número total de observaciones marcadas como *positivas*. Es decir, cuán acertado estuvo el algoritmo al afirmar que una observación era positiva.
- *Recall* (exhaustividad), que mide cuántas observaciones identificadas como *positivas* se correspondían al número real de observaciones *positivas*. Es decir, qué porcentaje de las observaciones positivas pudo detectar el algoritmo.
- *F-score* (Valor-F), que es la media armónica de *precision* y *recall*.

Notemos que al medir la *performance* de los algoritmos, es importante que el conjunto de ejemplos utilizado al evaluar sea disjunto con el usado para entrenar. De esta forma, la evaluación se hará siempre sobre ejemplos que no fueron vistos durante el entrenamiento. Para esto es común realizar una partición de los corpus en tres partes disjuntas. El porcentaje suele variar, pero una división clásica es: un 80 % de los datos en un conjunto para el entrenamiento (*training*), un 10 % en un conjunto para realizar ajustes de mejora de *performance* durante el desarrollo (*development*) y el 10 % restante para la evaluación final (*test*).

Para finalizar esta sección, observemos que la noción de corpus también puede extenderse a problemas donde se necesitan dos o más textos con cierta correspondencia. En la siguiente sección definiremos el problema de la traducción automática, donde este tipo de corpus cumple un rol fundamental.

2.3. Traducción Automática

La traducción automática es una subárea del PLN que se ocupa de traducir automáticamente texto de un lenguaje origen a otro lenguaje destino utilizando métodos computacionales. Por lo tanto, para trabajar en este problema, se necesitan colecciones de textos que cubran el uso del lenguaje origen y del lenguaje destino. A este tipo de corpus los denominamos **corpus paralelos** (Jurafsky y Martin, 2021) y están compuestos por pares de textos que se corresponden en una y otra lengua, donde la correspondencia entre ellos puede darse a nivel de documento, de oración o de palabra. El proceso de definir esta correspondencia se conoce como *alineación*.

Existen diferentes factores que hacen que el problema de traducción automática sea un problema difícil (Garg y Agarwal, 2019), como por ejemplo:

- no todas las palabras en un lenguaje tienen un equivalente en el otro
- dos lenguajes pueden diferir completamente en su estructura
- existen palabras con más de un significado

Por otra parte, el hecho de que uno de los lenguajes involucrados en la traducción sea *low resource* —como es el caso del guaraní— incrementa la dificultad del problema.

Por último, destacamos dos propiedades que definen la calidad de una traducción: adecuación y fluidez. La adecuación describe el grado en que la traducción captura el significado exacto del texto original, mientras que la fluidez representa qué tan natural suena la traducción en el lenguaje destino.
El objetivo de los sistemas de traducción automática es entonces generar traducciones que maximicen estas dos medidas simultáneamente (Jurafsky y Martin, 2021).

2.3.1. Enfoques basados en reglas

Históricamente, los primeros enfoques para resolver el problema de la traducción automática fueron los basados en reglas. Estos enfoques pueden categorizarse en tres modelos generales (Jurafsky y Martin, 2021):

- **Traducción directa**
  La traducción directa consiste en traducir cada palabra del texto origen al lenguaje destino, mediante el uso de un diccionario bilingüe.
- **Transferencia**
  Los enfoques basados en transferencia comienzan construyendo un árbol sintáctico del texto de origen y, mediante aplicación de reglas, lo transforman en un árbol sintáctico del lenguaje destino. A partir del análisis sintáctico del lenguaje destino se genera la oración traducida.
- **Interlingua**
  Para el caso de interlingua se transforma el texto origen en una representación abstracta de su significado, llamada interlingua. Luego se genera la oración destino en base a la interlingua.

2.3.2. Enfoques estadísticos

La traducción automática estadística, propuesta inicialmente por Brown et al. (1990), define la probabilidad $P(T|S)$ como la probabilidad de que la oración T sea la traducción en el lenguaje destino de la oración S en el lenguaje origen. En base a esto define el problema de la traducción automática como:

$$\widetilde{T} = argmax_T \ P(T|S) \tag{2.1a}$$

$$\overset{Bayes}{=} argmax_T \ P(T)P(S|T) \tag{2.1b}$$

El modelo de lenguaje se encarga de calcular la probabilidad $P(T)$ de la ecuación 2.1b, determinando qué oraciones son fluidas en el lenguaje destino. Sus parámetros son estimados en base a un corpus monolingüe en el lenguaje destino.

Por otra parte el modelo de traducción calcula la probabilidad $P(S|T)$, midiendo la correspondencia léxica entre lenguajes. Sus parámetros se estiman

en base a un corpus paralelo formado por pares de oraciones lenguaje origen-lenguaje destino.
Finalmente existe otro componente llamado decodificador, que se encarga de acotar la búsqueda de la oración T que maximice la probabilidad planteada, de forma de que sea computacionalmente eficiente.

2.3.3. Enfoques basados en redes neuronales

Un enfoque más moderno para la traducción automática estadística es el basado en redes neuronales, propuesto inicialmente por Kalchbrenner y Blunsom (2013) y Sutskever et al. (2014).

Como ya mencionamos, el problema de la traducción automática puede definirse en términos probabilísticos según la ecuación 2.1a. Este enfoque propone usar modelos de redes neuronales para aprender esta distribución condicional de probabilidad en base a grandes corpus paralelos de entrenamiento. Una vez que el modelo aprende la distribución, la traducción puede ser generada buscando la oración en el lenguaje destino que maximice dicha probabilidad condicional (Bahdanau et al. 2014).

Los modelos de redes neuronales utilizados para traducción automática suelen estar compuestos por dos módulos denominados *encoder* y *decoder*. El *encoder* extrae una representación vectorial de largo fijo a partir de una oración de entrada (oración de largo variable en el lenguaje origen). Posteriormente el *decoder* toma como entrada la codificación de la oración origen y genera la oración traducida en el lenguaje destino (Cho et al. 2014).

Siendo aún un enfoque muy nuevo, las redes neuronales aplicadas a la traducción automática han demostrado resultados prometedores. En algunos dominios y pares de lenguajes han superado a los sistemas estadísticos clásicos (Bahdanau et al. 2014; Castilho et al. 2017).

En este proyecto utilizamos OpenNMT (*Open-Source Toolkit for Neural Machine Translation*) (Klein et al. 2017), un framework *open source* que provee implementaciones basadas en redes neuronales para varias tareas de traducción automática. En particular utilizamos la configuración por defecto que

consiste en una arquitectura *encoder-decoder*, donde tanto el *encoder* como el *decoder* son una red neuronal recurrente (*RNN*) de tipo *LSTM*. Además, utiliza *atención* sobre la secuencia de entrada e implementa *input feeding* (Luong et al. 2015)[1].

2.3.4. Evaluación automática de traducciones

La evaluación humana de traducciones es indiscutiblemente la más exacta, pero es una tarea muy costosa. Los sistemas de traducción automática necesitan ser evaluados iterativamente durante su desarrollo, por lo que en la mayoría de los casos resulta inviable que estas evaluaciones se hagan de forma manual. Para resolver este problema se suelen utilizar métodos de evaluación automática.

Existen diversos métodos de evaluación automática de traducciones. A continuación se describen dos: *BLEU* (*bilingual evaluation understudy*), que es uno de lo más populares (Jurafsky y Martin, 2021) y CHRF, que fue utilizado como medida principal para generar los rankings oficiales del *shared task* de AmericasNLP, sobre traducción automática de lenguas nativas americanas (Mager et al. 2021).

2.3.4.1. BLEU

BLEU (*bilingual evaluation understudy*) es un método automático de evaluación de traducción automática. Este método se diseñó con el objetivo de ser de bajo costo computacional y de generar una métrica de evaluación que tenga alta correlación con los criterios del juicio humano, independientemente del lenguaje (Papineni et al. 2002).

El método recibe como entrada un conjunto de candidatos (traducciones que se quieren evaluar) y, para cada candidato, una o más referencias (traducciones correctas). Para calcular la métrica en un conjunto de oraciones se

[1] https://opennmt.net/OpenNMT/training/models/ - Accedido por última vez el 10.09.2021.

comienza por calcular lo que denominan una *precisión modificada*:

$$p_n = \frac{\sum_{C \in Candidates} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in Candidates} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

donde $Count_{clip} = min(Count, Max\_Ref\_Count)$.

Es decir, el numerador se calcula sumando, por cada candidato, cuántos de sus n-gramas ocurren en alguna de sus traducciones de referencia. El denominador es la suma del conteo de n-gramas de cada candidato.

El conteo del numerador se "recorta" en caso de que un candidato repita muchas veces un n-grama presente en la referencia, para evitar que las traducciones obtengan puntajes altos por el simple hecho de repetir n-gramas. Es por esto que se le llama precisión **modificada** de n-gramas.

Para ilustrar el concepto anterior, en Papineni et al. 2002 se presenta el siguiente ejemplo: dada la referencia "the cat is on the mat", un sistema de traducción automática genera el candidato "the the the the the the the". Este candidato, a pesar de ser una traducción pésima, obtiene una precisión estándar de unigramas de 7/7, mientras que su precisión modificada es de 2/7.

Por otra parte se introduce un coeficiente llamado *brevity penalty* (BP) que penaliza candidatos cuyo largo es menor al de las referencias. Se define como:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{\frac{1-r}{c}} & \text{if } c \leq r \end{cases}$$

donde c es el largo del corpus de candidatos (suma del largo de los candidatos) y r es el largo efectivo del corpus de referencias. Para calcular r es necesario elegir una de las K referencias que tenga asociadas un candidato, esto se hace tomando la referencia cuyo largo sea más cercano al del candidato.

Finalmente la métrica *BLEU* se calcula como:

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log p_n)$$

donde N indica el orden máximo de n-gramas a utilizar y los pesos w_i son valores positivos tales que $\sum_{i=1}^{N} w_i = 1$.

En este proyecto utilizamos la implementación de BLEU provista por la biblioteca NLTK[1] (Bird et al. 2009), usando los parámetros por defecto $N = 4$ y $w_i = 1/4$. Además utilizamos la función *Smoothing 1* descrita por Chen y Cherry (2014) para lidiar con los casos en que la precisión modificada es cero para algún de orden de n-gramas.

2.3.4.2. CHRF

La métrica *CHRF* se calcula como un *F-score* utilizando n-gramas de caracteres. Es de costo computacional bajo, independiente del lenguaje y no requiere de una etapa de *tokenización* (Popović, 2015). Esta medida ha demostrado también tener alta correlación con evaluaciones humanas, especialmente en lenguajes morfológicamente ricos (Popović, 2017).

La fórmula general de *CHRF* se define como:

$$CHRF\beta = (1 + \beta^2) \frac{CHRP \cdot CHRR}{\beta^2 \cdot CHRP + CHRR}$$

donde CHRP y CHRR representan el promedio de la precisión y *recall* sobre todos los n-gramas, respectivamente.

Es decir, CHRP representa el porcentaje de n-gramas en la hipótesis (predicción del modelo) que tienen un correspondiente en la referencia, y CHRR el porcentaje n-gramas en la referencia que también están presentes en la hipótesis. El conteo se realiza teniendo en cuenta solo una vez cada coincidencia de n-gramas. Para ilustrar el caso anterior, consideremos el ejemplo presentado

[1]NLTK es una biblioteca disponible para Python que provee varias funcionalidades para trabajar en PLN.

en la sección 2.3.4.1; allí el trigrama *the* del candidato será tenido en cuenta solo dos veces, una por cada vez que aparece en la oración de referencia.

Por su parte, el parámetro β permite dar mayor peso a una medida o la otra (precisión o *recall*). En particular si $\beta = 1$ ambos tienen el mismo peso, en general se le da β veces más peso al *recall* que a la precisión.

En este proyecto utilizamos la implementación de ChrF provista por la biblioteca NLTK (Bird et al. 2009), usando los parámetros por defecto $\beta = 3$, considerando desde unigramas hasta n-gramas de orden seis.

2.4. Vectores de palabras

Los vectores de palabras (*word embeddings*) son un modelo de semántica distribucional, que surge de tomar como hipótesis que las palabras que ocurren en los mismos contextos suelen tener significados similares (Lenci, 2018). Lo que buscan es construir representaciones de palabras mediante vectores densos (con pocas entradas en cero), intentando mantener la propiedad de que palabras "similares" tengan representaciones "cercanas" en el espacio. Por ejemplo, en la figura 2.2 podemos observar que palabras con una connotación similar sobre *disgusto* aparecen juntas en rojo mientras que lo análogo sucede para palabras que expresan *gusto*, en verde.

Figura 2.2: Distribución en el espacio para vectores de dimensión 60, extraído de Jurafsky y Martin (2021).

Por esta razón es que los vectores son muy utilizados para reforzar aspectos semánticos en tareas de PLN, ya sea análisis de sentimientos (Çano y Morisio, 2019), cálculo de similitud de documentos (Kusner et al. 2015) o traducción automática (Qi et al. 2018), entre muchos otros.

Existen muchas formas de construir estos vectores de palabras. Actualmente, la distinción más grande se hace entre vectores **estáticos** y **dinámicos**. Los vectores estáticos pretenden construir un único vector por cada palabra en el vocabulario, mientras que los dinámicos tienen en cuenta los diferentes contextos en los que aparece. Esto genera que para cada palabra en cada contexto hay un vector diferente (Jurafsky y Martin, 2021).

Entre los métodos para construir vectores estáticos se encuentran *fastext* (Bojanowski et al. 2017), *GloVe* (Pennington et al. 2014) y *Word2Vec* (Mikolov, Chen et al. 2013). En este trabajo construiremos vectores estáticos de Word2Vec utilizando el algoritmo *c-bow*; otro posible algoritmo para construirlos es *skip-gram*.

En cuanto a los vectores dinámicos, los métodos más conocidos son ELMO (Peters et al. 2018) y BERT (Devlin et al. 2019). Este último ha superado el estado del arte en muchas tareas de PLN (Rogers et al. 2020).

2.4.1. Evaluación

Para evaluar la calidad de los vectores de palabras hay diversos métodos que, en general, se pueden clasificar en tests **extrínsecos** e **intrínsecos** (Wang et al. 2019).

Los tests extrínsecos buscan evaluar los vectores de palabras en otras tareas de PLN, por ejemplo enriqueciendo un modelo de análisis de sentimientos, un POS-Tagger o ayudando a planear líneas de diálogo en sistemas conversacionales. La lógica detrás de estos tests es que las colecciones de vectores de buena calidad deberían tender a mejorar la *performance* de las tareas, pues se contaría con información léxica extra, que a priori no se tenía. En nuestro proyecto medimos la calidad de los vectores de esta forma al usarlos en los experimentos de traducción automática, como detallaremos en el Capítulo 5.

Por otra parte, los tests intrínsecos se centran en probar la calidad de los vectores de manera aislada, independientemente de cuánto puedan influir en la *performance* de tareas particulares. Entre ellos se encuentran los tests de **analogías** y los de **similitud**, ambos usados en este proyecto. Este test busca medir si la colección de vectores logra capturar las relaciones de analogías de manera similar a lo que podría hacer un humano. Por ejemplo, si tratamos de pensar en la analogía "París es a Francia como Montevideo es a...", seguramente tendamos a completar la frase con "Uruguay". Formalmente, el test se basa en predecir la palabra d que completa la cuaterna $< a, b, c, d >$, donde:

- a y b tienen cierta relación.
- d debe ser tal que cumpla la misma relación con respecto a c.

Así que para el ejemplo de París, Francia y Montevideo lo que se busca predecir es:

- a =París
- b =Francia
- c =Montevideo
- d =Uruguay

En general estos tests se calculan mediante el *vector-offset method* (Mikolov, Yih et al. 2013), que intenta encontrar el vector más cercano según la distancia coseno. Para ello primero se expresa al vector de la palabra buscada d como $y = vector_b - vector_a + vector_c$. El vector y resultante de la operación puede o no coincidir con alguno perteneciente a la colección de vectores. El problema se transforma entonces en hallar un vector w que sí pertenezca y que maximice su similitud de coseno con y:

$$w = argmax_w \frac{vector_w . y}{\|vector_w\| \|y\|}$$

Existen varios conjuntos para realizar estos tests. En nuestro proyecto usamos un subconjunto traducido de la colección original presentada por Mikolov, Chen et al. (2013)[1], lo que detallaremos en la sección 4.2.

[1] https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art) - Accedido por última vez el 22.08.2021.

2.4.2. Test de Similitud

Por otra parte, el test de similitud intenta medir la capacidad de capturar la percepción que tenemos los humanos sobre el parecido de dos conceptos del mundo. Por ejemplo ¿Cuánto se asemeja una *silla* a una *mesa*? ¿Y una *guitarra* a un *bosque*?

Estos tests no son tan simples de diseñar como los de analogías pues no solo basta con elegir pares de palabras, sino que hay que hallar también un valor de similitud esperado para ellos, típicamente en el intervalo $[0, 10]$. Uno de los enfoques usados para hallar estos valores es usando anotadores humanos (Jurafsky y Martin, 2021) con suficiente manejo de la lengua de modo de poder distinguir el significado de las palabras más allá del contexto, como es el caso de *SimLex-999* (Hill et al. 2015).

En cuanto a la computación, hay varias formas de calcular la similitud entre dos palabras. Para los vectores de palabras, la forma más usada es calculando la similitud del coseno entre los vectores que representan a las palabras del test (Wang et al. 2019). Luego de calcular la similitud entre todos los pares de palabras del test, se calcula la *correlación de Spearman* entre las medidas de similitud obtenidas y las esperadas, lo que da un valor perteneciente a $[-1, 1]$. Un resultado cercano a 1 indica que la correlación es prácticamente exacta, entorno al 0 que las observaciones no presentan correlación y tendiendo a -1 que la correlación es opuesta.

Finalmente existen varios conjuntos de tests de similitud[1], como el ya mencionado *SimLex-999*, *WordSimilarity-353* (Finkelstein et al. 2002) o, el que usamos en este proyecto, *MC-30* (Miller y Charles, 1991).

[1] https://aclweb.org/aclwiki/Similarity_(State_of_the_art) - Accedido por última vez el 22.08.2021.

2.5. PLN para el Guaraní

A pesar de ser una lengua con un número importante de hablantes, la investigación en PLN para el guaraní o jopará no es tan abundante como podría esperarse. Ha habido varios esfuerzos de trabajar en la lengua pero, como mostraremos en el desarrollo de este informe, la falta de recursos sobre los que apoyarse es un claro obstáculo.

El corpus de referencia para el Guaraní de Paraguay se denomina CO-REGUAPA (Secretaría de Políticas Lingüísticas del Paraguay, 2019)[1]. Si bien puede ser consultado mediante palabras o expresiones y sus resultados muestran las ocurrencias de los términos en el contexto, el corpus no puede ser descargado en su totalidad, lo que dificulta su uso para tareas de PLN.
Ha habido otros esfuerzos por vencer el obstáculo de la falta de datos, ya sea descargando texto de Twitter (Agüero-Torales et al. 2021; Ríos et al. 2014) o construyendo un pequeño corpus de *Universal Dependencies* del dialecto hablado por los Mbya (Dooley, 2006; Thomas, 2019), una parte de la población guaraní originaria. A este último trabajo no lo pudimos tener en cuenta pues el dialecto Mbya es bastante diferente al jopará, en el que nos enfocaremos en este proyecto.

Por último se cuenta con un pequeño corpus paralelo español-guaraní usado en la *shared task* de traducción automática en *AmericasNLP*[2] (Mager et al. 2021), un *workshop* de lenguas indígenas de las américas que tuvo lugar junto a la NAACL 2021. Para la competencia se realizó una traducción del corpus multilingüe XNLI (Conneau et al. 2018) a todas las lenguas de la *shared task*, con el objetivo de utilizarlo como evaluación y aprovechando a fortalecer el volumen de texto paralelo disponible para las lenguas indígenas. La naturaleza del texto en él está explicado con gran detalle en el artículo original, pero lo más relevante para nuestro proyecto es que hay muchos diálogos transcriptos, lo que lo diferencia del tipo de texto típicamente disponible para las lenguas de escasos recursos. Esta traducción también se hizo para el guaraní y cuenta con aproximadamente 1.000 oraciones para el conjunto de *development* y 1.000 para el conjunto de *test*.

[1] http://corpus.spl.gov.py/ - Accedido por última vez el 22.08.2021.
[2] http://turing.iimas.unam.mx/americasnlp/index.html - Accedido por última vez el 22.08.2021.

En lo que respecta a la resolución de tareas de PLN, existen trabajos en análisis morfológico (Kuznetsova y Tyers, 2021) y de traducción automática, que toman en consideración la falta de datos para la lengua (Abdelali et al. 2006; Alcaraz y Alcaraz, 2020; Gasser, 2018; Rudnick et al. 2014). Sobre traducción automática también se tiene registro de todos los enfoques probados por los participantes del *shared task* de AmericasNLP que participaron en la competencia de traducción Español-Guaraní (Bollmann et al. 2021; Knowles et al. 2021; Nagoudi et al. 2021; Parida et al. 2021; Vázquez et al. 2021; Zheng et al. 2021).

Para nuestro proyecto uno de los trabajos previos más relevantes es el corpus presentado por Chiruzzo et al. (2020). Este corpus paralelo guaraní-español fue extraído de sitios web de noticias y blogs paraguayos. Cuenta con aproximadamente 14.500 pares de oraciones, 228.000 *tokens* en guaraní y 336.000 *tokens* en español. A su vez, este trabajo fue reforzado por el de Borges y Mercant (2019). En él, las autoras exploran el uso de análisis morfológico para potenciar la *performance* en experimentos de traducción automática. En nuestra investigación usamos varios de sus resultados, por ejemplo la partición en conjuntos de *training*, *development* y *test* del corpus paralelo (Chiruzzo et al. 2020).

Capítulo 3

Construcción del Corpus

Uno de los objetivos de este proyecto fue ampliar la cantidad de texto disponible, tanto con texto paralelo como monolingüe. Considerando la condición de escasos recursos del guaraní, esta etapa del proyecto resultó fundamental para el desarrollo de los experimentos presentados y analizados más adelante.

En este capítulo detallamos entonces el proceso de construcción del corpus resultante, conformado por un conjunto paralelo guaraní-español de sitios de noticias y otro monolingüe compuesto de *tweets*.

3.1. Conjunto paralelo

Al iniciar el proyecto, el conjunto paralelo disponible para trabajar era el presentado en Chiruzzo et al. 2020[1], que fue construido mediante consultas manuales a portales de noticias. Uno de los aspectos trabajados fue la automatización de ese proceso para lograr conseguir al menos esa misma cantidad de texto paralelo.

En esta sección describiremos el proceso de construcción del conjunto paralelo que, como veremos en el capítulo 5, fue una de las piezas clave en el desempeño de los modelos de traducción.

[1]El conjunto paralelo usado en la *shared task* de AmericasNLP, descrito en la sección 2.5, fue publicado recién entre enero y marzo del 2021.

3.1.1. Crawling general en sitios de Paraguay

Inicialmente implementamos un *crawler* utilizando la biblioteca Scrapy para descargar contenido de varios sitios pertenecientes al *TLD* de Paraguay (.py). Para conseguir las *URLs* iniciales realizamos una serie de búsquedas automatizadas en Google, construyendo *queries* a partir de una lista de palabras frecuentes presentada en Borges y Mercant, 2019.

El *crawler* parte de estas *URLs* iniciales y continúa navegando por los diferentes enlaces para descargar más contenido. Para acotar la cantidad de sitios descargados el *crawler* sigue enlaces que pertenecen al mismo dominio y no descarga más de 50 páginas por dominio. Luego de descargar los sitios web realizamos un procesamiento para eliminar tanto etiquetas *HTML* como contenido no relevante (anuncios, menús, etc.).

En este primer experimento se recolectó texto de unas 6.700 páginas, y se tomaron 100 al azar para realizar un *sanity check*. Estas 100 páginas estaban mayormente en español, y solo 7 tenían unas pocas palabras en guaraní (mayormente nombres propios). Este resultado es coherente lo reportado en H. Jauhiainen et al. 2020, donde se plantea la dificultad de construir corpus basados en sitios web para lenguajes con poca representación en la web, incluso buscando en dominios pertenecientes al *TLD* del país en el que más se habla esta lengua (en nuestro caso, Paraguay).

Por lo tanto, el texto extraído en este experimento inicial fue descartado por no cumplir con un mínimo de calidad para el objetivo planteado. Sin embargo, mediante la revisión manual de las páginas extraídas, se encontraron dos sitios[1] [2] de noticias que publican consistentemente contenido tanto en guaraní como en español. Por lo tanto estos dos sitios sí fueron utilizados como punto de partida para la construcción del conjunto paralelo de textos.

3.1.2. Crawling particular en sitios de noticias

Con la certeza de que los dos sitios hallados publican regularmente contenido paralelo, decidimos automatizar la descarga, limpieza y alineación (ver

[1] https://www.abc.com.py/ - Accedido por última vez el 22.08.2021.
[2] http://www.spl.gov.py/ - Accedido por última vez el 22.08.2021.

sección 2.3) para ellos. Construimos entonces dos *crawlers* que descargan y procesan las noticias obteniendo su título, fecha de publicación y cuerpo, entre otros atributos. Inicialmente realizamos una descarga masiva para obtener los artículos ya publicados y luego configuramos un *cron job* para que diariamente se ejecutaran los *crawlers* y descargaran nuevas noticias. El total de artículos descargados fueron publicados entre octubre de 2020 y abril de 2021.

En cuanto al primer sitio, cada artículo en guaraní contaba con un enlace hacia su versión en español. El proceso de *crawling* comienza descargando el artículo en guaraní para luego navegar hacia el artículo en español, descargarlo y asociar ambas versiones.

Sin embargo el segundo sitio no contaba con una asociación explícita entre las versiones en guaraní y español de sus artículos, por lo que fue necesario implementar una heurística que deduzca esta relación. Los datos descargados hasta el momento mostraban que las versiones en guaraní y español de un mismo artículo eran generalmente publicadas el mismo día con aproximadamente treinta minutos de diferencia. En base a esta observación la heurística agrupa los artículos por fecha de creación, y relaciona cada uno con el más cercano de su grupo en hora de creación. Es decir, si hay cuatro artículos para el 31 de julio, dos en español y dos en guaraní, con horas de creación 14:34, 14:54, 17:27 y 17:57, la heurística agrupa a los primeros dos como versiones de un mismo artículo y a los últimos dos como versiones de otro artículo.

A pesar de que el patrón de los horarios se cumple muy seguido, existen algunos casos en los no puede determinar la relación:

- Si en una misma fecha la cantidad de artículos en guaraní no coincide con la cantidad de artículos en español.
- Si un artículo en guaraní es asociado con dos artículos en español, por compartir la hora de publicación más cercana (o viceversa).

En cualquiera de estos casos la heurística es robusta y logra marcarlos para revisión. La gran mayoría de los artículos de este sitio fueron asociados automáticamente y solo un pequeño porcentaje requirió revisión manual.

Finalmente la heurística se evaluó empíricamente mediante un chequeo manual de 100 pares de artículos, obteniendo un 100 % de resultados correctos.

3.1.3. Conjunto final

Como se indicó en la parte anterior, en base al contenido publicado en dos sitios de noticias paraguayos construimos un conjunto paralelo español-guaraní. El conjunto de artículos fue *tokenizado* a nivel de oración utilizando la función `sent_tokenize` del módulo `tokenize` que forma parte de la biblioteca *NLTK*. La alineación se realizó a nivel de oración utilizando una heurística basada en n-gramas de caracteres[1] descrita en Chiruzzo et al. 2020. En la tabla 3.1 se muestran ejemplos de pares de oraciones luego de ser alineadas.

gn	Oguahêvo 9:15 oñepyrû oguata hikuái mbo'ehárakuéra.
es	Cerca de las 9:15 se inició la marcha docente.
gn	Oñemopu'ãma paro total UNA-pe
es	Levantan paro total en la UNA
gn	Fiscalia ojerúre policia-pe ome'êvo proteccion umi periodista ha ifamilia-pe
es	Fiscalía pide a la Policía dar protección a periodista y su familia
gn	Comunicador-kuéra indígena omomba'e yvyramáta okáipagui ka'aguy
es	Comunicadores indígenas destacan valor del árbol ante incendios forestales

Tabla 3.1: Ejemplos de pares de oraciones extraídos del conjunto presentado.

El corpus paralelo construido en Chiruzzo et al. 2020 contó con un proceso de revisión manual, en el que se corrigieron errores tanto de *tokenizado* como de alineación de las oraciones. Allí se reporta que la versión manualmente revisada y corregida (enfoque semi-automático) tiene una calidad superior al enfoque completamente automático. Para la construcción de nuestro corpus se utilizó el enfoque automático, lo que implica que la calidad de sus pares de oraciones sea probablemente menor al presentado en Chiruzzo et al. 2020. Sin embargo, uniendo estos dos corpus se logra duplicar la cantidad de texto paralelo español-guaraní, lo que creemos puede mejorar la *performance* en tareas como la traducción automática. En el capítulo 5 veremos que esta hipótesis se confirma.

[1] Secuencias de n caracteres.

Con el fin de utilizar este corpus en los experimentos de traducción automática (que serán desarrollados en el Capítulo 5), se lo particióno en tres conjuntos: *training* (80 %), *development* (10 %) y *test* (10 %). Esta partición se hizo de forma aleatoria a nivel de documento, lo que nos garantiza que las oraciones de un mismo documento estarán en el mismo conjunto. El *script* que genera la partición utiliza una semilla de aleatoriedad fija, asegurando que sea replicable.

La primera versión del corpus paralelo fue construida en marzo de 2021 y reportada en Góngora et al. 2021. En abril de 2021 se generó otra versión que incluye algunos documentos más que la anterior y es la utilizada en los experimentos presentados en este proyecto. En la tabla 3.2 se muestran algunas estadísticas generales comparando las dos versiones recién mencionadas (Marzo 2021 y Abril 2021).

	Marzo 2021	Abril 2021
Documentos	2.580	2.652
Oraciones	14.792	15.175
Tokens en guaraní	334.497	343.565
Tokens en español	635.226	662.763

Tabla 3.2: Estadísticas del corpus presentado en Góngora et al. 2021 (Marzo 2021) y el utilizado en los experimentos de este proyecto (Abril 2021).

3.2. Conjunto de tweets

El lenguaje empleado en las redes sociales difiere mucho del que se puede encontrar en textos periodísticos, religiosos o académicos. Uno de los fenómenos lingüísticos presentes es el llamado *code switching* (cambio de código) que consiste en alternar la lengua usada al expresarse (Jurafsky y Martin, 2021), cada vez más frecuente debido al diario intercambio cultural producido en internet. Otras diferencias son, por ejemplo, la alteración de las reglas gramaticales o la inclusión de símbolos como emojis o letras de alfabetos de otras lenguas.

Para agregar otro tipo de texto al corpus en construcción, considerando las características mencionadas y aprovechando que la red social *Twitter* dispone

de una API de fácil acceso, nos propusimos construir un conjunto de *tweets* para el jopará. Como vimos en la sección 2.1, el jopará presenta diferentes niveles de mezcla con el español, por lo que esperábamos que esta condición se mantuviera o, incluso, se intensificara.

Ya que a priori no conocíamos el grado de mezcla y *code switching* presente en los *tweets*, decidimos capturar tantos candidatos como fuera posible para realizar, posteriormente, un filtrado con ciertas restricciones. Como veremos al final de esta sección, conseguimos una gran variedad de *tweets* que, mediante una heurística, fueron clasificados según su grado de mezcla.

3.2.1. Obtención de *tweets* a partir de palabras clave

La API de *Twitter*[1] dispone de varias funcionalidades que permiten, entre otras cosas, obtener *tweets* en tiempo real. La funcionalidad que más se adaptaba a nuestro caso era la de la búsqueda mediante una *query* de palabras clave[2].

Por lo tanto, para construir el conjunto de *tweets*, se consultó periódicamente[3] a la API utilizando como *query* algunas de las palabras frecuentes en guaraní[4]. Estas palabras fueron obtenidas de la lista construida por Borges y Mercant (2019), usada también para la construcción del conjunto paralelo descrito en la sección 3.1. En un principio intentamos recoger texto solamente dentro del territorio paraguayo[5], pero nos encontramos con dos problemas:

1. Estábamos dejando afuera muchos *tweets* en guaraní que no tenían disponible la información sobre su geolocalización. Es decir, no eran capturados debido a nuestra restricción de admitir solo *tweets* dentro del territorio paraguayo.
2. Muchos de los *tweets* recuperados por la *query* no estaban en guaraní, sino en español o portugués.

[1] https://developer.twitter.com/en/docs/twitter-api/v1 - Accedido por última vez el 22.08.2021.

[2] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets - Accedido por última vez el 22.08.2021.

[3] Con una frecuencia de entre 3 y 8 minutos.

[4] En cada consulta se toma una muestra aleatoria de quince palabras.

[5] Restringido mediante el rango de coordenadas de Paraguay.

Sobre el primer problema decidimos comenzar a recolectar *tweets* tanto del territorio paraguayo (lo denominaremos *local*) como de cualquier parte del mundo (lo denominaremos *global*). Esto complejizó mucho más el segundo problema encontrado, ya que a partir de eso comenzamos a tener interferencia de muchas otras lenguas. Por esa razón vimos la necesidad de tener algún método o herramienta para determinar si los *tweets* presentaban contenido en guaraní o no.

3.2.2. Reconocimiento de guaraní

Como mencionamos, el primer problema que tuvimos al obtener una gran cantidad de *tweets* — a partir de una *query* con palabras frecuentes — fue que no todos ellos estaban en guaraní, por lo que necesitábamos alguna manera de distinguirlos del resto. Si bien la API de Twitter ya intenta identificar el lenguaje de los *tweets* enviados como respuesta[1], en todos los experimentos que realizamos no logramos obtener ningún caso donde se etiquetara un *tweet* como guaraní, incluso en aquellos donde el contenido estaba completamente en guaraní.

El primer enfoque para resolver este problema consistió en utilizar un identificador de lengua estadístico.

3.2.2.1. Reconocimiento estadístico

Realizamos experimentos utilizando dos identificadores de lengua estadísticos. Uno de ellos es *Polyglot* (Al-Rfou' et al. 2013), que es muy usado en la comunidad ya que tiene herramientas para algunas lenguas de escasos recursos, entre ellas el guaraní. Esta herramienta con el módulo *language detection* (detección de lengua) que permite predecir cuál o cuáles son las lenguas más probables en las que está escrito un texto[2]. El otro es un clasificador tipo Naive Bayes[3] que entrenamos sobre el conjunto de *training* de Chiruzzo et al. (2020), basándonos en T. Jauhiainen et al. 2018.

[1] Campo *lang* de la respuesta de la API.

[2] https://polyglot.readthedocs.io/en/latest/Detection.html - Accedido por última vez el 22.08.2021.

[3] Clasificador *GuassianNaiveBayes* entrenado con *features* equivalentes a los 5-gramas de caracteres más frecuentes del conjunto de entrenamiento, utilizando la biblioteca *Scikit-learn*.

Para la evaluación de estas herramientas se tomó todo el conjunto de *test* de Chiruzzo et al. 2020 y, para cada oración, se intentó identificar si correspondía a una oración en guaraní o no. Tengamos en cuenta que al ser un conjunto paralelo guaraní-español las oraciones de cada lengua están alineadas y claramente identificadas.

El desempeño de nuestro clasificador fue sorprendente, obteniendo 99,58 % en la medida F1. En la tabla 3.3 se puede ver su resultado en la evaluación y también el obtenido por el identificador de *Polyglot*.

Herramienta	**Precision**	**Recall**	**F1**
Polyglot	96,90	96,90	96,90
NaiveBayes+5-gramBoW	99,58	99,58	99,58

Tabla 3.3: *Performance* de las herramientas para clasificar correctamente oraciones en guaraní. La evaluación fue realizada sobre el conjunto de test de Chiruzzo et al. 2020.

El excelente resultado de nuestro clasificador está dado por el estilo de escritura de la partición de *test*, que se corresponde a la misma del conjunto de *training* usado para entrenarlo. Por su parte *Polyglot* funcionó muy bien aunque estuvo unos puntos por debajo. Tengamos en cuenta que, al estar entrenado para identificar diversas lenguas, su evaluación tuvo que adaptarse. Mientras que nuestro clasificador cuenta con dos salidas posibles (`es` o `gn`), *Polyglot* devuelve para cada lengua la probabilidad de que el texto esté escrito en ella. Por lo tanto, se decidió:

- tomar como salida de *Polyglot* la lengua más probable. Es decir, si el guaraní tenía un 65 % de probabilidad y el portugués un 20 %, se tomaba como salida a "guaraní".
- si la salida era cualquier otra lengua que no fuera guaraní, lo contábamos como "español". Recordemos que en el conjunto de evaluación (Test2020) solamente hay oraciones en español y guaraní.

De esta forma logramos comparar la salida de nuestro clasificador y la de *Polyglot*.

A pesar de los buenos resultados en la evaluación, su *performance* no se mantuvo al probarlos sobre los *tweets*, ya que ambos clasificadores identificaban guaraní en muchos *tweets* que no lo tenían. Por un lado, recordemos que nuestro clasificador estaba entrenado sobre el texto periodístico de Chiruzzo et al. 2020, el cual posee cierta estructura para mantener su formalidad, que difiere mucho del estilo de escritura de las redes sociales. Por otra parte, al revisar los *tweets* que eran erróneamente identificados como escritos en guaraní, vimos que habían algunos en japonés, en español, en portugués, en inglés, en francés, en tailandés y en turco. Esta diversidad de idiomas afectó de gran manera el desempeño del clasificador, que solamente fue entrenado para desambiguar entre el español y el guaraní. Por su parte *Polyglot* sí estaba entrenado para desambiguar entre más lenguas, pero tampoco lograba acertar en gran cantidad de casos.

Tengamos también en cuenta que estos clasificadores no están pensados para usar en contextos donde el *code switching* está tan presente. Por ejemplo, si evaluamos el siguiente *tweet* para decidir en qué lengua está:

> "Seguile está siguiendo a todos los que le siguen he'i,
> mbae pío cheve si me sigue o no ese sin deprovecho"

podría decirse que está escrito en español o guaraní. Sin embargo, para nosotros ese es un *tweet* en jopará, pues corresponde al grado de mezcla "español con palabras prestadas del guaraní", presentado en la sección 2.1.

Como no nos era posible curar manualmente el filtrado de *tweets* realizado por estos identificadores estadísticos, optamos por crear una heurística que, aunque dejara muchos *tweets* fuera, nos diera más seguridad de que los conservados tenían las características buscadas.

3.2.2.2. Reconocimiento por reglas

Ya que el reconocimiento estadístico no funcionó debido a la falta de texto apropiado para el entrenamiento, decidimos usar entonces un método basado en listas de palabras. Previo a diseñar la heurística, curamos manualmente la lista de palabras que estábamos usando para construir la *query*, quitando:

- Signos de puntuación
- Fechas
- Palabras que fueran usadas en otros idiomas. Por ejemplo, la palabra "táva" significa pueblo o aldea en guaraní, pero también se usa "tava" en portugués como una conjugación posible del verbo "estar". Otros ejemplos son "chupe", "lorenzo", "rire" ("reír" en francés), etc.

La lista resultante apunta a tener palabras que sean usadas casi exclusivamente en guaraní. A partir de ella, creamos dos sublistas:

- **Lista larga**: Conformada por las palabras de la lista curada que tienen 3 caracteres o más. Por ejemplo, la palabra "ha" no pertenece a esta lista pero "nde" sí. Esta lista tiene 314 palabras en total.
- **Lista corta**: Conformada por las palabras de la lista larga que cumplen tener una frecuencia igual o mayor a 11. Esta lista cuenta con 48 palabras en total.

Comenzamos entonces a descargar *tweets* que tuvieran alguna palabra de la lista corta. Luego incorporamos la lista larga con el objetivo de realizar una serie de chequeos que nos dieran alguna pista acerca de cómo decidir si los *tweets* conseguidos contaban con las características buscadas. Para eso tomamos una muestra aleatoria de 100 *tweets* de cada posible combinación de geolocalización (*tweet local* o *global*) y cantidad de palabras de la lista larga, y marcamos para cada uno si lo considerábamos en jopará o no. Por ejemplo, tomamos cien *tweets globales* aleatorios con dos palabras de la lista larga y evaluamos, uno por uno, si estaban o no en jopará. Los resultados de estos cuatro chequeos, realizados con algunas semanas de diferencia entre sí y por tanto con diferentes muestras, se pueden ver promediados en la tabla 3.4.

Geo. \Palabras de lista larga	**1**	**2**	**3**	**4**	**5**
Local	92 %	100 %	100 %	100 %	100 %
Global	1 %	18 %	79 %	100 %	100 %

Tabla 3.4: Promedio de resultados de los cuatro chequeos manuales para detectar si los *tweets* eran considerados en jopará o no.

Nos sorprendió gratamente que ciertas combinaciones — de geolocalización y cantidad de palabras — lograran tener *tweets* en jopará en todos nuestros muestreos. La estadística generada nos permitió definir empíricamente una serie de restricciones mínimas para que el *tweet* fuera incluido en nuestra colección:

- Si el *tweet* es *global* debe de tener al menos 3 palabras frecuentes de la lista larga.
- Si el *tweet* es *local* debe de tener al menos 1 palabra frecuente de la lista larga. Sin embargo, encontramos tres palabras que son particularmente ruidosas: “ary”, “ara” y “guasu”. Un *tweet* que **solamente** tenga una de esas palabras no será considerado en jopará.

En el algoritmo 1 se resume el método utilizado para la recolección de *tweets*.

Algoritmo 1 Heurística de filtrado de *tweets* basada en lista de palabras.

$listaLarga \leftarrow \{palabra : palabra \in listaCurada \wedge largo(palabra) \geq 3\}$
$listaCorta \leftarrow \{palabra : palabra \in listaLarga \wedge frecuencia(palabra) \geq 11\}$
$palabrasQuery \leftarrow obtenerAleatorio(listaCorta, 15)$
$query \leftarrow$ "w_1 OR w_2 OR ... OR w_{15}", $w_i \in palabrasQuery$
$tweetsCandidatos \leftarrow obtenerTweets(query)$
$tweetsFinales \leftarrow []$
for each $tweet \in tweetsCandidatos$ **do**
  $palabras\_tweet\_larga = palabras(tweet) \cap listaLarga$
  if $esGlobal(tweet)$ AND $largo(palabras\_tweet\_larga) \geq 3$ **then**
    $tweetsFinales.insert(tweet)$
  else if $esLocal(tweet)$ AND $largo(palabras\_tweet\_larga) > 1$ **then**
    $tweetsFinales.insert(tweet)$
  else if $esLocal(tweet)$ AND $largo(palabras\_tweet\_larga) = 1$ **then**
    if $palabras\_tweet\_larga \not\subset$ {"ary", "ara', "guasu"} **then**
      $tweetsFinales.insert(tweet)$
    end if
  end if
end for
return $tweetsFinales$

3.2.3. Conjunto final

En la sección anterior vimos que realizamos cuatro chequeos manuales, a lo largo de varias semanas, con el objetivo de obtener algunas restricciones mínimas que tienen que cumplir los *tweets* para ser incluidos en nuestra colección. Observemos que para los *tweets* locales con 2 o más palabras o los globales con 4 o más, los chequeos manuales nos indican que hay mucha seguridad de que el *tweet* esté en jopará. Sin embargo esta seguridad disminuye si alguna de esas condiciones no se cumple. A raíz de esto y aprovechando los patrones observados, clasificamos cada tweet en A, B o C:

- **A**: Son *tweets globales* con 4 palabras frecuentes o más, o *tweets locales* con 3 palabras frecuentes o más. Estos *tweets* tienen un alto contenido de expresiones en guaraní. Muchos de ellos están, incluso, escritos completamente en guaraní.

- **B**: Son *tweets locales* con 2 palabras frecuentes. En esta categoría se aprecia fuertemente la mezcla de español y guaraní. La presencia del guaraní se manifiesta en palabras aisladas o pequeñas expresiones.
- **C**: Son *tweets globales* con 3 palabras frecuentes, o *tweets locales* con 1 palabra frecuente. Esta categoría contiene a todos los *tweets* que no pueden ser automáticamente desambiguados. El método empleado con la lista de palabras no es lo suficientemente robusto para cubrir estos casos donde, muchas veces, hay palabras en guaraní poco frecuentes. Por esa razón es que en esta categoría hay muchos *tweets* en guaraní, pero también en portugués, filipino o español.

De acuerdo a la definición anterior y los resultados observables en la tabla 3.4, las categorías A y B nos dan — empíricamente — 100 % de seguridad de que sus *tweets* se corresponden a textos en jopará, por lo que los denominamos *tweets confiables*. En la tabla 3.6 se pueden ver ejemplos para cada una de las categorías, donde se observan las características descritas.

Por último, en la tabla 3.5 se muestra el total de *tokens* para tres versiones del conjunto de tweets:

- Marzo 2021: versión reportada en Góngora et al. 2021.
- Junio 2021: versión utilizada para los experimentos finales de este proyecto.
- Agosto 2021: versión definitiva al finalizar el proyecto.

Es decir que para la última versión de este conjunto (Agosto2021) logramos obtener 9.365 *tweets confiables*, que suman un total de 111.471 *tokens*.

Categoría	**Marzo 2021**		**Junio 2021**		**Agosto 2021**	
	Tweets	*Tokens*	*Tweets*	*Tokens*	*Tweets*	*Tokens*
A	532	7.706	811	11.791	1.009	14.734
B	4.199	48.895	6.498	75.493	8.356	96.737
C	46.197	453.996	71.767	706.907	87.633	867.210
Total	50.928	510.597	79.076	794.191	96.998	978.681

Tabla 3.5: Evolución de la cantidad de *tweets* y tokens a través de los últimos meses del proyecto. A los *tweets* de las categorías A y B los llamamos *tweets confiables* debido al fuerte contenido de guaraní presente en ellos.

Cat.	Tweet
A	@GustavoVelazque Ko'a nio o pagapata en vida ko yvy apere... Ha'e oimo'a ho'u reita... Ojereta hese si o si.
A	@Arandu_juky @lthxfender JAJAJA nde suerte katu, ha'e ndoikuaa moaî mba'e la ñande ja'e
A	@elposerotaku Ha maa piko he'i ndeve que estoy en asunción nde añamemby metido... kampaña ruguare aî nde tavyron
A	@GretaThunberg Moopio nde reikuaa mbaeve mitakuñai revy kya. Terehona pekaru nde poha mbae tavy. Best wished to you.
A	Que belleza. Guãnguīngué, ha'e opuraheiramo pyharé malaguero he'i ñandeve ñande abuelita kuera. Añetepiko pea.
B	como pio queres que te haga caso si ni mi historia respondes nde mba'e tavy
B	@MatthewHedges Kóa pio noñemondóima hóga lado ra'e ko oshutáva hi'árko ládo? Ha'e péa pe témandevoi oikuaa. Havõ nde pýre. Go your home!
B	No te soporto pero che tavyete ave nde rehe nde añamemby plaga
B	Nde pukavy ombohory che rekove
B	Mi hermano se recibió ayer y mi familia ya me reclama que uno de mi promo también ya se va a recibir. Ha mbae pio cheve peñe calma!
B	Yo vine a aprender a fluir hei mi carta astral. Y bueno, vamos a fluir mbae xq sino nervia!!!!
C	@claribernal Por que tuiteas y no me respondes nde sabandija?
C	Una patrullera chocada por un poste mbae jeyma la oikoa
C	Para que pico vas a hacerle caso a la gente ñe'ereí, en esta vida nde evy'ante arã y que te chupe un huevo lo que digan los demás
C	Iporã laja nde membyyyy señora, ijapuete reiiiiiii ndeko
C	Iñaroooooo la ñati'u nde barbaro, en jeans ko estoy e igual nomás che su'u vaipaiteee ko añarako, caradura ko son
C	Una persona tatuada da mala impresión he'i che Dios ñande jukata ko mentalidad retrógrada
C	Nadie Adsolutamente nadie Yo cuando alguién me mira: "Mba'ei areko pío nde tele she rovare"
C	@spillthespells Ary aryyy asy ksy ana pra hmry bulany b to aty Nhi
C	@WalterEvers Ndee que aburrido todo, y si nos cepillamos los dientes mbae.
C	Nde eporandu chupe?
C	@thierrison @malheirostattoo @JuninhoAmarilla @silsilveiraa Saca tua vez kkkkkk sem bololo em gordinho do karai ahaushhshshs
C	Que persona del bien es Ary, es increíble la cantidad de consejos y cosas linda que me dice

Tabla 3.6: Ejemplos de los *tweets* que conforman cada una de las categorías.

3.3. Otros textos

Para algunos de los experimentos, utilizamos además otros conjuntos monolingües en base a contenido disponible en la web que tuvimos que procesar y limpiar:

- **Wikipedia**: Descargamos un *dump* de la Wikipedia en guaraní[1] y realizamos un posprocesamiento para eliminar contenido no relevante y extraer solo el texto de los artículos.
  Este conjunto está formado por un total de 6.123 artículos[2] y 504.730 *tokens*.
- **Libro de Mormón**: Implementamos un proceso automático para extraer la versión oficial en guaraní del Libro de Mormón, publicada en el sitio de La Iglesia de Jesucristo de los Santos de los Últimos Días[3].
  Este conjunto cuenta con 243 documentos y 204.434 *tokens*.
- **La Biblia**: Implementamos un proceso de *crawling* que permitió extraer dos versiones de La Biblia publicadas en la web[4]:
  - *Ñandeyara Ñe'ẽ* [5], que corresponde a una traducción del Antiguo Testamento y del Nuevo Testamento de la Biblia católica.
  - *Tûpâ Ñandeyára 1913*[6], que corresponde a una traducción de la *King James Bible*.

  Ambas versiones suman un total de 1.451 documentos y 760.697 *tokens*.

[1] https://dumps.wikimedia.org/gnwiki/ - Dump correspondiente al 20.02.2021.
[2] Tengamos en cuenta que la mayoría de estos artículos tienen menos contenido que las correspondientes páginas en otras lenguas, como el español o el inglés.
[3] https://www.churchofjesuschrist.org/study/scriptures/bofm?lang=grn - Descargado el 09.07.2021.
[4] https://biblics.com/gn - Descargado el 11.07.2021.
[5] https://www.bible.com/versions/2315 - Accedido por última vez el 22.08.2021.
[6] https://www.bible.com/versions/805 - Accedido por última vez el 22.08.2021.

Capítulo 4

Vectores de Palabras

Si bien nuestro objetivo principal era desarrollar recursos para fortalecer la traducción automática entre el español y el guaraní, también experimentamos con la construcción de vectores de palabras sobre el texto monolingüe en guaraní. Como desarrollamos en la sección 2.4, los vectores de palabras son una de las posibles formas de representar la semántica léxica y tienen muchísimas aplicaciones en la resolución de tareas de PLN. Por ejemplo, pueden ser usados para hallar un valor de similitud entre dos documentos u oraciones y resolver una parte de un sistema de preguntas y respuestas automático.

La mayor parte del trabajo presentado en este capítulo fue guiado por los experimentos presentados en Etcheverry y Wonsever, 2016 y lo resultados iniciales fueron presentados en Góngora et al. 2021.

4.1. Entrenamiento

A lo largo del proyecto se entrenaron alrededor de 300 modelos con el objetivo de probar diferentes hiperparámetros y colecciones de textos; varias de estas configuraciones se reportan en Góngora et al. 2021. A partir de todos los experimentos iniciales realizados, elegimos un conjunto de 24 configuraciones relevantes para entrenar, evaluar y detallar en este informe.

Todos estos modelos fueron construidos usando la implementación del algoritmo *c-bow* para *Word2Vec* de la herramienta Gensim (Řehůřek y Sojka, 2010). El texto utilizado en el entrenamiento se detalla en la tabla 4.1 y está

conformado por todos los textos detallados en el capítulo 3. El preprocesamiento consistió en:

- Normalizar los apóstrofos usados por un único símbolo (').
  Esto se debe a que los usuarios que generan estos textos suelen usar diferentes símbolos para expresar el *puso*, consonante del alfabeto guaraní descrita en la sección 2.1.
- Eliminar espacios en blanco redundantes.
- Transformar todo el texto a minúsculas.
- *Tokenizar* usando el `tweet_tokenizer` de NLTK.

Conjunto	**Tokens**
Biblia	760.697
Libro del Mormón	204.434
Wikipedia	504.730
Chiruzzo et al. (2020) + Conjunto paralelo 2021	433.134
Subtotal - Texto usado para **todos** los modelos	1.902.995
tweets - Categoría A	11.791
tweets - Categoría B	75.493
tweets - Categoría C	706.907
Total	2.697.186

Tabla 4.1: Conjuntos de textos usados y sus cantidades de *tokens*. El conjunto paralelo se corresponde a la versión de **abril** presentada en la tabla 3.2 y los *tweets* usados a la versión correspondiente a **junio** presentada en la tabla 3.5.

Como se muestra en la tabla, todos los modelos usan la parte guaraní de los conjuntos paralelos, la *Biblia*, el *Libro de Mormón* y la *Wikipedia*.
Si bien los 24 modelos se entrenaron durante 60 iteraciones sobre esos conjuntos y utilizaron todos los *tokens* disponibles, aunque aparecieran solamente una vez en el texto, hay ciertas particularidades que los diferencia entre sí, y son combinaciones de:

- **Dimensión de los vectores**: 150 y 300.
- **Tamaño de la ventana**[1]: 6, 7 y 9.
- **Categorías de *tweets* usadas en el entrenamiento**: sin *tweets*, categoría A, categorías A y B, y categorías A, B y C.

[1] Al construir la representación vectorial para una palabra del vocabulario se toma en consideración el contexto en el que aparece. La ventana determina la cantidad de *tokens* que se toman en consideración en la oración, hacia un lado y otro de la ocurrencia de la palabra.

Según las categorías de *tweets* usadas para el entrenamiento, el tamaño del vocabulario resultante para las 24 colecciones varía entre 151.092 y 197.741 palabras.

A continuación detallaremos el proceso de evaluación de estos modelos, que comenzó con la creación o traducción de algunos de los tests comunmente usados (Wang et al. 2019). Posteriormente analizaremos varios aspectos, como la diversidad léxica presente en los modelos o la interferencia introducida por el característico *code switching* de los *tweets*.

4.2. Tests de analogías y similitud

Para medir la calidad de las colecciones nos basamos en los experimentos realizados por Etcheverry y Wonsever (2016), consistentes en **tests de analogías** de palabras (ver sección 2.4.1), **tests de similitud** de palabras (ver sección 2.4.2) y una **visualización** mediante reducción de dimensionalidad para ver cómo se agrupan las palabras en el espacio vectorial resultante.

Para poder realizar los tests mencionados primero tuvimos que realizar una traducción al guaraní, o construir un conjunto equivalente al original pero pensado desde el comienzo en guaraní. Al no contar con herramientas de traducción automática, el proceso de traducción al guaraní fue manual y por lo tanto —como no somos hablantes nativos— extenso y costoso. Por esta razón, además del experimento de visualización, decidimos concentrarnos en solo tres tests intrínsecos que describiremos a continuación: **family**, **capital-common-countries** y **MC-30**.

4.2.1. Test de analogías: Family

Este test de analogías, propuesto por Mikolov, Chen et al. (2013), trata sobre relaciones familiares y sus variaciones utilizando los géneros masculino y femenino. Por ejemplo, una cuaterna del test original es: $\langle boy, girl, brother, sister \rangle$.

Este test no pudo ser traducido de manera directa desde el original. Por un lado, la limitante del conocimiento del idioma no nos permitió desambi-

guar ciertas definiciones en los diccionarios. Por otra parte, algunas palabras presentes en el test no existen en guaraní o sus traducciones eran la misma palabra en guaraní, lo cual le quita utilidad al test.

Lo que hicimos entonces fue construir un nuevo test para el guaraní basado en el original, utilizando los siguientes pares de palabras:

1. ⟨ tamói, jaryi ⟩ ≈ ⟨ abuelo, abuela ⟩
2. ⟨ ména, tembireko ⟩ ≈ ⟨ esposo, esposa ⟩
3. ⟨ túva, sy ⟩ ≈ ⟨ padre, madre ⟩
4. ⟨ tuvanga, syanga ⟩ ≈ ⟨ padrino, madrina⟩
5. ⟨ mitã'i, mitãkuña'i ⟩ ≈ ⟨ niño, niña ⟩
6. ⟨ kuimba'e, kuña ⟩ ≈ ⟨ hombre, mujer ⟩
7. ⟨ karai, kuñakarai ⟩ ≈ ⟨ señor, señora ⟩

Luego de cruzar todos los pares de palabras entre sí obtuvimos 42 cuaternas para realizar el test.

4.2.2. Test de analogías: Capital-Common-Countries

Este test de analogías, también propuesto por Mikolov, Chen et al. (2013), trata sobre ciudades capitales de países frecuentemente nombrados. Una cuaterna del test original es por ejemplo:

$$\langle Berlin, Germany, Madrid, Spain \rangle$$

Es importante aclarar que la hipótesis de este test (países *frecuentemente* nombrados) no tiene por qué cumplirse para los textos en guaraní con los que contamos. Por ejemplo, el único país americano presente es Cuba; no aparece ni Uruguay, ni Argentina, ni Paraguay. A pesar de esta condición decidimos traducirlo, ya que es uno de los conjuntos más utilizados al evaluar colecciones de vectores.

El test original pudo ser traducido en su integridad. Apoyándonos en la Wikipedia en guaraní, tradujimos cada nombre de ciudad y país utilizando el título de sus artículos. El test traducido cuenta de un total de 506 cuaternas.

4.2.3. Test de similitud: MC-30

El test de similitud MC-30, propuesto por Miller y Charles (1991), está compuesto por 30 pares de palabras puntuadas del 0 al 4 en función de su similitud. Por ejemplo, en el test se encuentra:

- $score(\langle magician, wizard \rangle) = 3,50$
- $score(\langle food, fruit \rangle) = 3,08$
- $score(\langle magician, glass \rangle) = 0,11$

El test fue traducido con la ayuda de diversos diccionarios online pero sobre todo gracias a la colaboración del hablante nativo Marvin Agüero-Torales. Del conjunto de palabras original, las únicas palabras que no pudieron ser traducidas fueron "asylum" (asilo) y "madhouse" (manicomio).

4.2.4. Resultados

En la tabla 4.2 presentamos los resultados de los tests de *analogías* y *similitud* para los 24 modelos entrenados . Se incluye una columna para cada característica relevante de configuración del modelo: tamaño de los vectores (*size*), tamaño de la ventana usada para evaluar el contexto (W) y conjunto de *tweets* usado[1]. Incluimos además una fila para los mejores resultados conseguidos a la fecha de publicación de nuestro paper (Góngora et al. 2021), una fila para los más altos (*Máximos*) y otra para los más bajos (*Mínimos*) resultados de toda la tabla presentada. La tabla se presenta ordenada por el resultado obtenido en el test MC-30, ya que hay varios autores que consideran que, de entre los tests intrínsecos, los tests de similitud son buenos para guiarse (Jurafsky y Martin, 2021).

[1]Si la celda está vacía significa que no se usaron *tweets* en el entrenamiento.

model_name	Size	W	Tweets	familyT1	familyT5	cccT1	cccT5	MC-30
Máximos	-	-	-	54,76	59,52	9,49	21,34	0,569
Mínimos	-	-	-	38,10	47,62	4,15	14,43	0,403
Paper	-	-	-	41,27	48,41	5,53	13,37	-
Vectores13	300	6		45,24	47,62	7,91	17,59	**0,569**
Vectores06	150	7	A	50,00	52,38	7,11	15,61	0,556
Vectores15	300	6	AB	40,48	50,00	5,93	17,00	0,552
Vectores23	300	9	AB	47,62	52,38	8,10	19,76	0,543
Vectores16	300	6	ABC	40,48	47,62	4,74	17,98	0,541
Vectores20	300	7	ABC	40,48	52,38	8,70	17,79	0,538
Vectores03	150	6	AB	42,86	52,38	7,71	18,77	0,530
Vectores02	150	6	A	45,24	57,14	7,11	17,39	0,527
Vectores22	300	9	A	45,24	57,14	7,71	18,38	0,521
Vectores21	300	9		50,00	54,76	6,52	17,59	0,519
Vectores01	150	6		42,86	52,38	6,52	18,58	0,515
Vectores24	300	9	ABC	38,10	54,76	6,32	20,16	0,513
Vectores19	300	7	AB	50,00	**59,52**	**9,49**	18,97	0,512
Vectores18	300	7	A	45,24	52,38	7,51	20,16	0,511
Vectores08	150	7	ABC	45,24	54,76	4,35	14,43	0,502
Vectores04	150	6	ABC	45,24	52,38	4,15	15,42	0,500
Vectores07	150	7	AB	40,48	54,76	8,10	18,38	0,499
Vectores09	150	9		45,24	54,76	9,09	**21,34**	0,495
Vectores10	150	9	A	45,24	54,76	6,92	18,38	0,475
Vectores14	300	6	A	42,86	54,76	8,10	17,79	0,473
Vectores12	150	9	ABC	42,86	52,38	6,52	19,17	0,460
Vectores11	150	9	AB	50,00	54,76	7,31	17,19	0,449
Vectores05	150	7		**54,76**	54,76	9,09	18,77	0,440
Vectores17	300	7		42,86	52,38	7,71	20,95	0,403

Tabla 4.2: Experimentos monolingües finales de modelos entrenados durante 60 épocas sobre diferentes configuraciones. Se muestran los valores máximos y mínimos de cada test en negrita y subrayados, respectivamente.

Los tests de analogías fueron evaluados mediante la obtención de la palabra más similar (*exact match*) y las cinco más similares (*top 5 match*)[1], para luego comprobar si la palabra esperada se encuentra en alguna de esas. Estas variaciones están indicadas en la tabla como "T1" y "T5" correspondientemente y su valor expresa el porcentaje de aciertos. El test de similitud se calculó como la correlación de Spearman entre los valores de similitud esperados y los hallados entre los vectores[2].

[1] Usando la función *most_similar* disponible en la biblioteca Gensim.
[2] Usando la función *similarity* provista también por Gensim.

Analizando los resultados vemos que, como era de esperar, todos los resultados de los tests de analogías son superiores al tomar 5 palabras para evaluarlos. Por otro lado, el uso de *tweets* o la dimensión de los vectores parecería no determinar la *performance* del modelo: hay muy buenos resultados con y sin *tweets*, así como los hay con dimensión 150 y 300. Sin embargo sí se observa que ningún modelo entrenado usando *tweets* de la categoría C llegó a obtener el máximo en algún test, al mismo tiempo que estos *tweets* sí se usaron en algunos modelos con el puntaje mínimo, como es el caso de `Vectores04`, `Vectores08`, `Vectores16` y `Vectores24`. Análogamente, vemos que los modelos que usan estrictamente las categorías A y B obtuvieron dos puntajes máximos y ningún puntaje mínimo. Esto puede indicar que los *tweets* de la categoría C, a pesar de tener buenos datos en algunos casos, tienden a ser en general más ruidosos y eso afecta la calidad de los modelos entrenados.
Por último, es claro que la inclusión de más texto permitió superar los resultados obtenidos en Góngora et al. 2021[1]. Un ejemplo de esto es el modelo `Vectores19` que logró superar todos los valores obtenidos en el paper además de conseguir dos valores máximos: en el test *familyT5* y en el *cccT1*.

En comparación, los resultados para el test *ccc* no fueron tan buenos como para el de *family* y creemos que esto se debe a la falta de diversidad en los textos usados para el entrenamiento. Mientras que la denominación para los integrantes de la familia puede aparecer normalmente en textos de noticias o bíblicos, no todos los nombres de países y ciudades de la actualidad suelen aparecer en ellos. Sería ideal contar con diferentes tipos de texto como cuentos, poemas, textos legales, manuales técnicos o hilos de foros de discusión, ya que esto podría brindar al modelo entrenado un mejor desempeño en diferentes escenarios. Tengamos en cuenta que el texto usado de los conjuntos paralelos es principalmente extraído de noticias, por lo que tienden a tener una cierta estructura y vocabulario que se repite consistentemente, mientras que los textos religiosos usan otras expresiones un poco más arcaicas y no cubren temas actuales de política o geografía. De las fuentes de textos usados para el entrenamiento, la *Wikipedia* quizás sea la que contenga más referencias a este

[1] Al momento de realizar los experimentos reportados en el paper contábamos con menos texto del conjunto de noticias y *tweets*. Además, no contábamos con La Biblia ni el Libro de Mormón, descritos en 3.3.

tipo de temas, pero como vimos en la tabla 4.1, representa una pequeña parte de los *tokens* usados para el entrenamiento.

Por último es importante recordar que el test *ccc* no toma en cuenta a ningún país de la región, lo cual inevitablmente impacta en el puntaje recibido, pues son los que tienen más probabilidad de aparecer en los textos usados para el entrenamiento. Como veremos en el siguiente capítulo, esta falta de diversidad de textos también se hará presente al evaluar los modelos de traducción automática.

En cuanto al test de similitud MC-30 los modelos no muestran resultados tan diferentes entre sí, pero se puede ver una leve diferencia entre el primero (`Vectores13`) y el último (`Vectores17`). En general los resultados fueron buenos ya que el más alto obtenido $(0,569)$ está cerca del más bajo reportado $(0,618)$ en el estado del arte detallado por la ACL[1]. Tengamos en cuenta que si bien allí hay reportados mejores valores, todos ellos refieren al estado del arte para el test en inglés, idioma que cuenta con gran variedad recursos desarrollados (Joshi et al. 2020).

Finalmente podemos observar que no hay una gran correlación entre los resultados de los tres tests. El modelo que resultó primero en MC-30, `Vectores13`, no cuenta con ningún máximo en otro test, pero sí es el mínimo para el test *familyT5*. Por otro lado, `Vectores05` resultó penúltimo en MC-30, pero tiene el valor máximo de *familyT1*. Por último `Vectores19`, que vimos que tenía dos valores máximos en los tests de analogías, quedó en la mitad de la tabla en cuanto a su resultado de MC-30.

[1] https://aclweb.org/aclwiki/MC-28_Test_Collection_(State_of_the_art) - Accedido por última vez el 22.08.2021.

4.3. Experimentos de visualización de palabras

Para el test de visualización nos basamos en el experimento realizado por Etcheverry y Wonsever (2016), donde se eligieron ciertas palabras del español buscando que pertenzcan a ciertas categorías, como fechas, animales, colores o personas. Posteriormente se graficaron, mediante una reducción de dimensionalidad, con el objetivo de observar cómo se agrupaban en el espacio según su significado.

Por lo tanto, tradujimos todas estas palabras y les asignamos un color a cada una de esas categorías para poder facilitar la visualización de los grupos formados al realizar el gráfico. En la tabla 4.3 se muestra la lista completa de palabras, con su correspondiente categoría semántica y el color asignado junto a la letra que lo representa[1]. La única palabra que no fue traducida[2] fue "violeta", la cual se sustituyó por *pytãngy*, que significa "rosado".

4.3.1. Resultados

Tomando la traducción realizada del experimento original (Etcheverry y Wonsever, 2016) y el color asociado a cada categoría semántica, graficamos en el espacio la posición de las diferentes palabras (descritas en la Tabla 4.3) mediante una reducción a dos dimensiones usando PCA (*Principal component analysis*) (Maćkiewicz y Ratajczak, 1993). Este experimento lo realizamos para varios de los 24 modelos entrenados, pero mostraremos aquí solamente aquellos que nos parecen interesantes de comentar.

[1]Cada categoría tiene además una letra que la identifica de modo que la lectura del experimento no dependa de los colores, tal como sugiere la guía de accesibilidad de la ACL al 22.08.2021.

[2]El texto disponible para entrenar los vectores, al momento de la traducción del test, no contenía ninguna ocurrencia de la palabra *violeta* en guaraní.

Español (original)	**Guaraní (traducción)**	**Categoría** y *color*	**Español (original)**	**Guaraní (traducción)**	**Categoría** y *color*
lunes	arakõi		bueno	porã	
martes	araapy		malo	vai	
miércoles	ararundy		feo	vai	
jueves	arapo	**día**	lindo	porã	
viernes	arapoteĩ	*negro* (k)	positivo	añetehápe	**atributo**
sábado	arapokõi		negativo	vai	*rojo* (r)
domingo	arateĩ		bondad	porã	
			belleza	porãngue	
			fealdad	vaikue	
enero	jasyteĩ		perú	perũ	
febrero	jasykõi		bolivia	volívia	
marzo	jasyapy		paraguay	paraguái	
abril	jasyrundy		uruguay	uruguái	
mayo	jasypo		argentina	argentina	**país**
junio	jasypoteĩ	**mes**	dinamarca	ndinamáka	*magenta* (m)
julio	jasypokõi	*negro* (k)	alemania	alemaña	
agosto	jasypoapy		holanda	olánda	
setiembre	jasyporundy		francia	hyãsia	
octubre	jasypa		italia	itália	
noviembre	jasypateĩ				
diciembre	jasypakõi				
1975	1975		**violeta**	**pytãngy**	
1897	1987		azul	hovy	
1998	1998	**año**	negro	hũ	**color**
1976	1976	*negro* (k)	rojo	pytã	*cian* (c)
1986	1986		blanco	morotĩ	
1992	1992				
delfín	piranare		juan	juan	
perro	jagua		fabiana	fabiana	
gato	mbarakaja		francisco	francisco	
tigre	jaguarete	**animal**	rodolfo	rodolfo	**personas**
loro	gua'a	*verde* (g)	roberto	roberto	*amarillo* (y)
foca	kyjachuroto		hector	héctor	
			juana	juana	
			laura	laura	
			romina	romina	

Tabla 4.3: Palabras traducidas al guaraní del test original en español (Etcheverry y Wonsever, 2016).

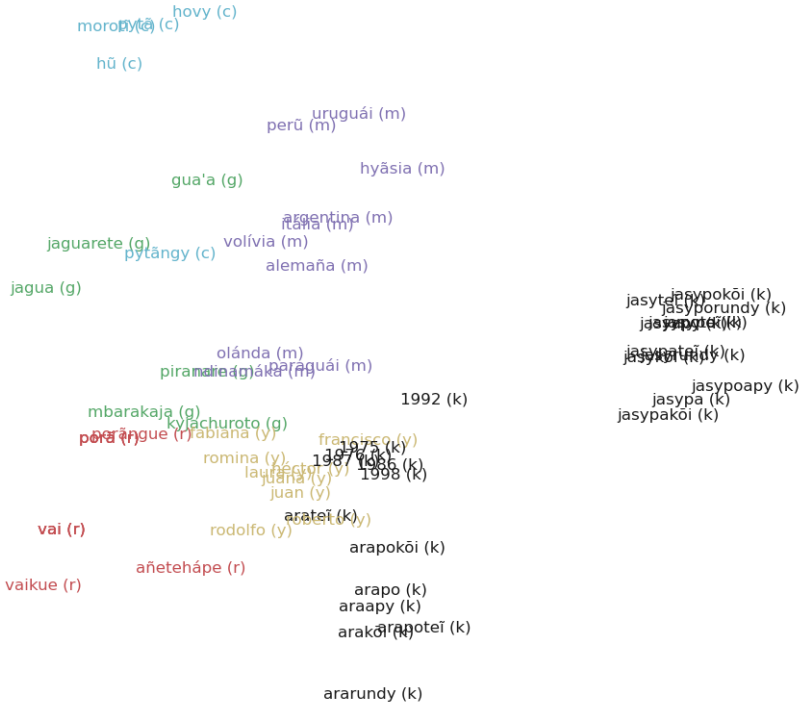
En primer lugar vamos a comparar los resultados mostrados en la figura 4.1, para dos de los mejores modelos según la tabla 4.2:

- `Vectores13`, que resultó primero según MC-30.
- `Vectores20`, que usando el conjunto de *tweets* en su integridad, consiguió muy buenos valores en todos los tests.

Observando las figuras 4.1a y 4.1b vemos que, aunque hay ciertas palabras que cambian levemente su posición, no hay grandes diferencias entre ambas. Sin embargo, podemos ver que en la figura 4.1a las palabras de la categoría en color rojo están más aisladas del resto que en la figura 4.1b. Por otra parte, en esta última se cumple que la palabra "francisco" está bastante más cerca de "argentina" que para el resto de modelos. Pensamos que esta cercanía puede darse debido a que el actual papa católico, llamado Jorge Bergoglio y que adoptó el nombre de "Francisco", es de nacionalidad argentina. También se puede ver en este gráfico lo cerca que se encuentran las palabras "jagua" (*perro*) y "mbarakaja" (*gato*).

Por otra parte vimos muy interesante realizar el experimento para `Vectores19`, que no quedó en los primeros lugares según el test de similitud pero sí consiguió obtener de los mejores puntajes en los tests de analogías. Analizando el resultado notamos que la palabra "francisco" se encuentra cerca del año "1992" y luego de una breve búsqueda encontramos que en ese año se dió la ordenación episcopal del sacerdote Bergoglio. Ya que esta coincidencia nos llamó la atención, decidimos repetir el experimento para este modelo pero incluyendo otras fechas que son importantes para Francisco, como su fecha de nacimiento (1936), su ordenación sacerdotal (1969), su proclamación cardenalicia (2001) o su elección como actual papa (2013). En la figura 4.2 podemos ver el resultado de este experimento con las nuevas palabras. Sorprendentemente, su fecha de nacimiento aparece en el gráfico sumamente cerca de su nombre. En cuanto al resto de grupos, las palabras "vai" (*malo* o *feo*) y "vaikue"(*fealdad*) están cerca, así como sucede con "porã"(*bueno* o *lindo*) y "porãngue" (*belleza*).

Finalmente es notorio que para los distintos modelos vistos, los colores, países, meses, días de la semana y años se muestran agrupados. Esto parece confirmar la intuición de que los vectores logran capturar información sobre el contexto en que las distintas palabras aparecen.



(a) Vectores13

(b) Vectores20

Figura 4.1: Visualización de cómo se agrupan las palabras para los modelos *Vectores13* y *Vectores20*, mediante una reducción de dimensionalidad usando PCA.

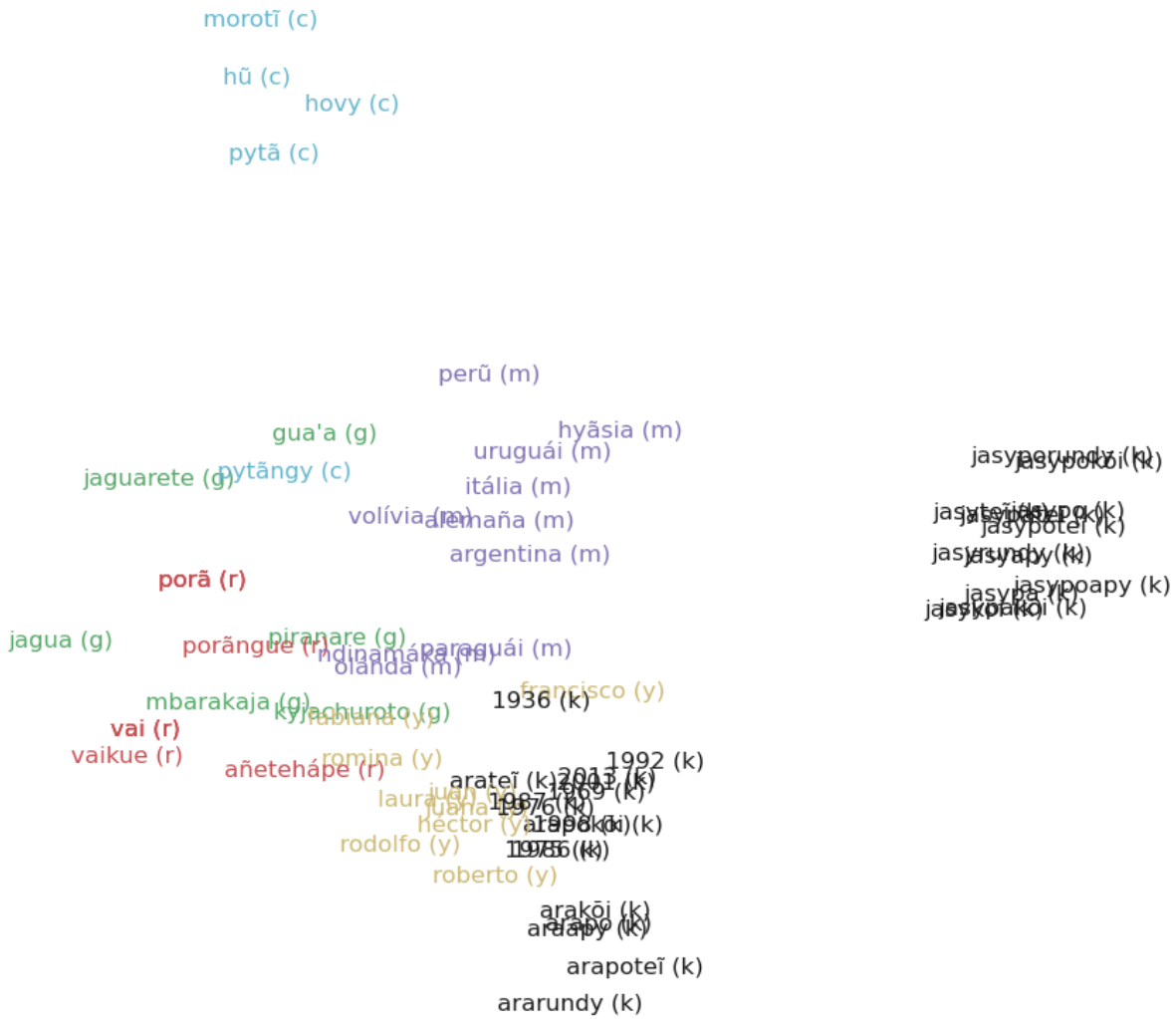


Figura 4.2: Gráfico para el modelo *Vectores19*.

Capítulo 5

Traducción automática

Como comentábamos en la introducción, el objetivo principal de este proyecto fue construir recursos para fortalecer la *performance* de la traducción automática entre el guaraní y el español. Usando los recursos construidos y detallados en los capítulos previos, construimos diversos modelos de traducción automática en ambos sentidos: guaraní-español y español-guaraní. Posteriormente evaluamos y comparamos su *performance* con los resultados obtenidos por Borges et al. 2021 y por los participantes de la ya mencionada *shared task* de AmericasNLP[1]. Además del análisis numérico de los resultados, realizamos dos experimentos exploratorios: *back-translation* y análisis de *performance* en función del largo de entrada.

En este capítulo detallaremos los conjuntos de datos utilizados para entrenar y evaluar estos modelos, así como también el análisis de los experimentos realizados.

5.1. Construcción de modelos

Para la construcción de los modelos utilizamos la versión `2.1.2` de OpenNMT-py, implementación basada en PyTorch del proyecto OpenNMT (Klein et al. 2017), utilizando el cómputo de GPU brindado por la plataforma Google Colab. Cada modelo fue entrenado durante 80.000 *steps*, persistiendo en disco una versión (*checkpoint*) cada 5.000 *steps*.

[1] A la fecha de escritura de este documento estos resultados representan, según nuestro conocimiento, todos los existentes para la traducción automática entre español y guaraní. No incluimos comparaciones con otros experimentos de traducción ya que las medidas dependen de cada par de lenguas, incluso en contextos de escasos recursos.

El texto utilizado para el entrenamiento fue preprocesado como se describe en la sección 4.1, y se compone de:

- Conjunto de *training* de Chiruzzo et al. 2020, obtenido de la partición generada por Borges et al. (2021).
- Conjunto de *training* del corpus paralelo desarrollado en este proyecto y detallado en la sección 3.1.

En algunos experimentos se utilizaron colecciones preentrenadas de vectores de palabras, logrando así hacer uso del texto monolingüe recolectado, además del texto paralelo ya mencionado. Para el español se utilizó la colección de dimensión 300 presentada en Azzinnari y Martínez, 2016. Las colecciones de vectores usadas para el guaraní son `Vectores13`, `Vectores23` y `Vectores20`, previamente presentados en el capítulo 4. Esta elección se hizo considerando que:

- `Vectores13` presenta el puntaje máximo en el test MC-30 y no fue entrenado con texto de *tweets*.
- `Vectores23` tiene buenos puntajes en todos los tests y se entrenó usando los *tweets* de las categorías A y B (*tweets confiables*).
- `Vectores20` es, de los que se entrenó con las tres categorías de *tweets* (A, B y C), uno de los que mejor se desempeña en el test MC-30 y en los otros tests.
- Por restricción de OpenNMT, la dimensión de los vectores elegidos para el guaraní, debe ser igual a la de los vectores en español.

Por lo tanto, las colecciones de vectores elegidas para el guaraní son de dimensión 300, de forma de coincidir con la dimensión de la colección para el español (Azzinnari y Martínez, 2016). Además, todas ellas tienen un buen desempeño en los tests intrínsecos y utilizan las diferentes categorías de *tweets* (ver tabla 4.2).

Para el resto de los parámetros de entrenamiento se utilizaron los **valores por defecto** provistos por el *framework*[1].

[1] https://opennmt.net/OpenNMT-py/options/train.html - Accedido por última vez el 22.08.2021.

Se desarrollaron entonces ocho modelos de traducción automática, cuatro en la dirección español-guaraní y cuatro en la dirección opuesta, variando la colección de vectores utilizada para el guaraní, los cuales denominaremos de la siguiente manera:

- `Gn-Es` y `Es-Gn`
  Entrenados sin utilizar una colección de vectores preentrenada.

- `Gn-Es+Vectores13` y `Es-Gn+Vectores13`
  Entrenados usando la colección `Vectores13`.

- `Gn-Es+Vectores23` y `Es-Gn+Vectores23`
  Entrenados usando la colección `Vectores23`.

- `Gn-Es+Vectores20` y `Es-Gn+Vectores20`
  Entrenados usando la colección `Vectores20`.

5.2. Guaraní-español

El experimento de mayor prioridad fue el del sentido guaraní-español, de forma de comparar nuestros resultados con los previamente obtenidos por Borges et al. (2021). En complemento con esa evaluación realizaremos un análisis exploratorio de la calidad de las traducciones mediante la observación, lo que es posible por ser el español nuestra lengua materna.

Durante el entrenamiento de cada modelo, se guardaron *checkpoints* cada 5.000 *steps*. Luego analizamos el comportamiento de cada uno de estos *checkpoints* sobre nuestro conjunto de desarrollo, y elegimos el que tenga mejor *performance* con esos datos. La elección se basó en la medida ChrF, ya que parece ser particularmente buena para lenguas polisintéticas como el guaraní (Mager et al. 2021), y por consistencia se usó esta misma métrica tanto en el sentido guaraní-español como en el español-guaraní (que veremos en la sección 5.3).

El conjunto de desarrollo (en adelante Dev2020-2021) utilizado es la unión de los siguientes conjuntos:

- Conjunto de *development* del corpus paralelo presentado en Chiruzzo et al. 2020 (Dev2020)
- Conjunto de *development* del corpus paralelo construido en este proyecto (Dev2021)

En la figura 5.1 se muestran los valores de BLEU y ChrF obtenidos en el conjunto de desarrollo para los diferentes *checkpoints* de los modelos entrenados.

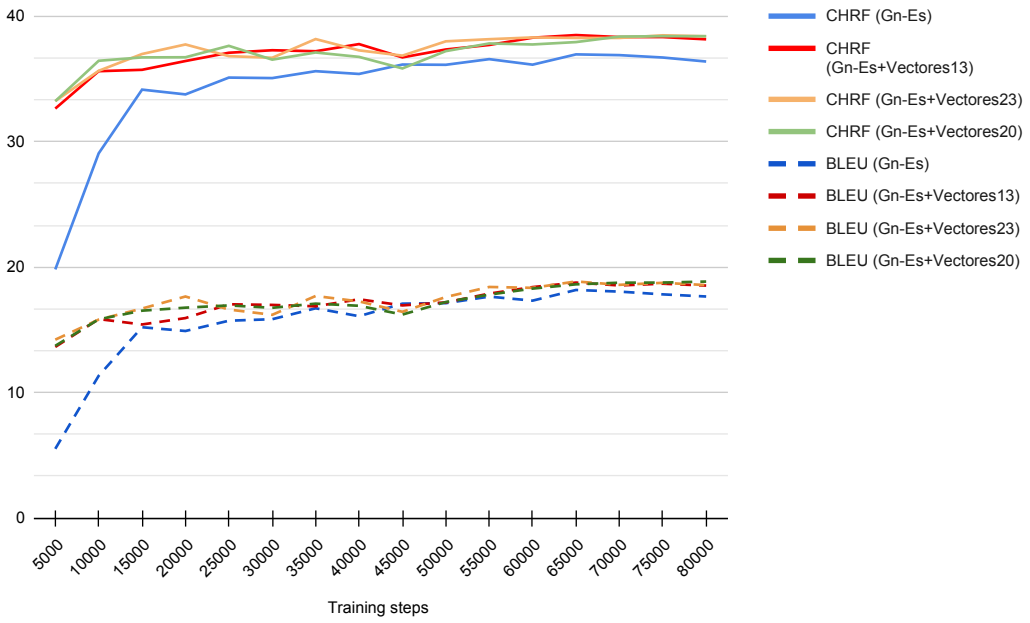


Figura 5.1: Evolución de BLEU y CHRF para los modelos guaraní-español sobre el corpus de desarrollo (Dev2020-2021).

Lo más relevante a observar es que los modelos que utilizan vectores comienzan a converger más tempranamente. Esto es especialmente beneficioso en el caso de no contar con suficiente poder o tiempo de cómputo para realizar largos entrenamientos, como fue nuestro caso. Por otra parte observamos que el modelo que no utiliza vectores está, en la mayoría de los *steps*, por debajo del que sí los usa. Finalmente vemos que luego de 40.000 *steps* el desempeño de los modelos no varía enormemente.

5.2.1. Resultados

Una vez obtenidos los modelos definitivos en la dirección guaraní-español según su desempeño en el conjunto de desarrollo, realizamos su evaluación sobre los conjuntos de test. Los conjuntos de evaluación utilizados fueron:

- Conjunto de *test* del corpus paralelo presentado en Chiruzzo et al. 2020 (Test2020)
- Conjunto de *test* del corpus paralelo construido en este proyecto (Test2021)
- Conjuntos de *development* (DevANLP) y *test* (TestANLP) utilizados en la *shared task* de Americas NLP (Mager et al. 2021)

Como métricas de evaluación automática utilizamos tanto BLEU como Chrf. En la tabla 5.1 se comparan los resultados reportados en Borges et al. 2021 con los obtenidos en este proyecto al evaluarlos sobre Test2020; tanto sus sistemas como los nuestros fueron desarrollados utilizando OpenNMT en su configuración por defecto. Desde el punto de vista de la medida BLEU consideramos que se logró una interesante mejora en la calidad de la traducción, especialmente teniendo en cuenta que nuestros modelos no hacen uso de ningún tipo de información morfológica. Podemos observar que el modelo sin vectores preentrenados (`Gn-Es`), cuya configuración coincide con la de los presentados en Borges et al. 2021, presenta visibles mejoras. Esto último indicaría que el principal factor de mejora en nuestros experimentos es el incremento en la cantidad de texto paralelo.

Modelo	**Test2020**	
	BLEU	ChrF
(Borges et al. 2021) - Exp 1	17,40	-
(Borges et al. 2021) - Exp 2	13,80	-
(Borges et al. 2021) - Exp 3	17,40	-
(Borges et al. 2021) - Exp 4	15,70	-
(Borges et al. 2021) - Exp 5	**20,30**	-
`Gn-Es`	21,90	37,26
`Gn-Es+Vectores13`	**22,64**	38,63
`Gn-Es+Vectores23`	22,49	38,32
`Gn-Es+Vectores20`	22,54	38,46

Tabla 5.1: Experimentos de traducción desde guaraní a español, evaluados sobre el conjunto Test2020.

En una vista más general, la tabla 5.2 presenta las medidas BLEU y ChrF obtenidas al evaluar los modelos sobre los conjuntos Test2020, Test2021, DevANLP y TestANLP. Como podemos ver, el modelo `Gn-Es+Vectores13` obtuvo el mejor desempeño en esta dirección al evaluarlo sobre los corpus Test2020 y Test2021. Creemos que esto puede estar relacionado al hecho de que el conjunto de vectores con el que fue entrenado (`Vectores13`) no incluyó *tweets* en su construcción. El texto de *tweets* suele ser más ruidoso que el de artículos periodísticos, siendo este último estilo de texto el presente en Test2020 y Test2021. También es notorio que la naturaleza del texto de AmericasNLP (descrito en 2.5) deteriora de gran manera la *performance* de nuestros modelos. Este problema viene de la mano de la diversidad de texto usado en el entrenamiento, que fue discutido en la sección 4.2.4 al evaluar los vectores de palabras construidos. Por último, volvemos a observar una mejora consistente comparando los modelos que usan vectores preentrenados contra los que no.

Modelo	**Step**	**Test2020**		**Test2021**		**DevANLP**		**TestANLP**	
		BLEU	ChrF	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF
`Gn-Es`	65.000	21,90	37,26	15,12	37,71	0,41	12,22	0,37	11,75
`Gn-Es+Vectores13`	65.000	22,64	**38,63**	15,75	**39,13**	0,48	13,44	0,51	12,85
`Gn-Es+Vectores23`	75.000	22,49	38,32	15,85	38,76	0,44	13,52	0,44	**12,93**
`Gn-Es+Vectores20`	80.000	22,54	38,46	15,75	38,94	0,57	**13,65**	0,50	12,75

Tabla 5.2: Experimentos de traducción desde guaraní a español, evaluados sobre Test2021, DevANLP y TestANLP.

5.2.2. Análisis de ejemplos de traducción

En la tabla 5.3 se observan algunos ejemplos de traducción de oraciones del conjunto Test2020 usando el modelo `Gn-Es+Vectores13`. Estos ejemplos se seleccionaron de forma manual con el objetivo de observar algunas características que creemos interesantes, aunque solo sea un análisis exploratorio no exhaustivo de la calidad de las traducciones generadas.

En el ejemplo #1 podemos notar que el texto de referencia en guaraní consta de dos oraciones, en vez de una. Esto probablemente se deba a un error en el proceso de *tokenización* o alineación de oraciones en el corpus del que proviene la oración (Test2020). Esto puede ser una de las causas por las que el ejemplo presenta baja adecuación y fluidez. Sin embargo se observa que el sujeto de la oración ("La Asociación de Empresarios Cristianos") es conservado correctamente, incluyendo el acrónimo ADEC. En varios de los ejemplos revisados se observa el patrón mencionado anteriormente: los nombres propios se suelen traducir correctamente independientemente de la calidad del resto de la traducción.

El ejemplo #2 presenta alta fluidez, y a menos de una variación en el tiempo verbal ("se está llevando a cabo" por "se realizó") la traducción generada conserva el significado de la frase original. En el ejemplo #3 también se muestra una gran fluidez y adecuación; el problema principal lo tiene en la concordancia de género entre el determinante y "estudio", además de omitir el día que dará lugar.

Nos resulta muy interesante el cambio que se realiza en la predicción del ejemplo #4. En la referencia se espera "las crecidas", mientras que la traducción usa " crecida del río" lo cual suena natural y es correcto desde el punto de vista del significado. Algo similar pasa para el ejemplo #6, donde se usa "en medio" en lugar de "en el centro", lo que a primera vista puede sonar equivalente pero al considerar el contexto en que se usa una y otra expresión no lo es. En la referencia, la expresión "en el centro" pretende dar la sensación de que el caso Messer es protagonista, lo cual al usar "en medio" parece que el caso hubiera pasado "durante" la agenda regional. Es una diferencia leve pero afecta a la adecuación de la traducción.

Por último en el ejemplo #8 se observa que, si bien la traducción generada es pobre, el modelo logra traducir correctamente los signos de puntuación (comillas, coma y punto final).

#	**Referencia (gn)**	**Referencia (es)**	**Predicción (es)**
1	Asociación de Empresarios Cristianos (ADEC) ome'ë jopói ko pyhare 11 pequeña ha grande empresa ojoajúva ha oñemoambuéva ojuehegui aporte social rehe. Presidente gremio-gua jkarai Juan Luis Ferreira, oñe'ë política ha tavaygua remiandu rehe.	La Asociación de Empresarios Cristianos (ADEC) premia esta noche a 11 pequeñas y grandes empresas que se diferencian por su aporte social.	La Asociación de Empresarios Cristianos (ADEC) es lo que comenzó con el 11 de las empresas y la empresa vinculada a la producción de sus aportes y la ilusión de que la población se encuentra en el país.
2	Ko ceremonia oñemotenonde instalaciones Banco Central del Paraguay (BCP).	La ceremonia se está llevando a cabo en las instalaciones del Banco Central del Paraguay (BCP).	La ceremonia se realizó en las instalaciones del Banco Central del Paraguay (BCP).
3	Ko estudio proyecto de ley de paridad democrática ofigura punto orden del día sesión ordinaria Senado-pe guarã juevesóúvape.	El estudio del proyecto de ley de paridad democrática figura en el punto del orden del día de la sesión ordinaria del Senado del próximo jueves.	La estudio del proyecto de ley de paridad democrática también será el punto de la orden del día en la sesión ordinaria del Senado.
4	FNC ojesolidarisa damnificadokuéra ndive crecida rehe	FNC se solidariza con damnificados por las crecidas	FNC se movilizará con damnificados por crecida del río
5	Orekorôguare 13 ary oikeva'ekue omba'apo Departamento Nacional de Trabajo-pe.	A los 13 años entró a trabajar en el Departamento Nacional del Trabajo.	A los 13 años de edad se logró trabajar en el Departamento Nacional de Trabajo.
6	Caso Messer agenda regional mbytépe	CASO MESSER EN EL CENTRO DE AGENDA REGIONAL	Caso Messer en medio de la agenda regional
7	Ogueromandu'ávo 100 ary relaciones diplomática ogueroguatáva Paraguay ha Japón, Gobernación de Itapúa oipepirû peteî serie de actividades ogueromandu'ávo ko joaju tetã oñondive.	Al cumplirse 100 años del establecimiento de las relaciones diplomáticas entre Paraguay y Japón, la Gobernación de Itapúa invita a una serie de actividades en homenaje a los lazos entre ambos países.	En recordación de más de 100 años de relaciones diplomáticas entre Paraguay y Japón, la Gobernación de Itapúa invita a una serie de actividades para celebrar la iniciativa.
8	"Péicha, ojeguereko omba'apóva coordinada Viceministerio de Tributación.	"De hecho, se tiene que trabajar de manera coordinada con el Viceministerio de Tributación.	"Por él, hay que hacer unidos el Viceministerio de Tributación.

Tabla 5.3: Ejemplos de traducciones del conjunto Test2020, generadas por el modelo Gn-Es+Vectores13.

5.3. Español-guaraní

Para elegir los modelos de traducción definitivos en el sentido español-guaraní, realizamos el mismo procedimiento que en el sentido opuesto. Es decir, se analiza la *performance* de cada *checkpoint* sobre el conjunto de desarrollo Dev2020-2021 y se elige el que mejor se desempeña según ChrF.

En la figura 5.2 vemos cómo evolucionan durante el entrenamiento las métricas de evaluación sobre el conjunto de desarrollo. En este caso los vectores de palabra no aseguran la convergencia más rápida como vimos para el sentido opuesto, ni tampoco aseguran obtener un mejor resultado a largo plazo. Los modelos `Es-Gn+Vectores23` y `Es-Gn+Vectores20`, ambos usando vectores de palabras, se encuentran por debajo de los otros dos durante casi todo el entrenamiento. Además, se puede observar que los valores comienzan oscilando considerablemente y lentamente tienden a converger.

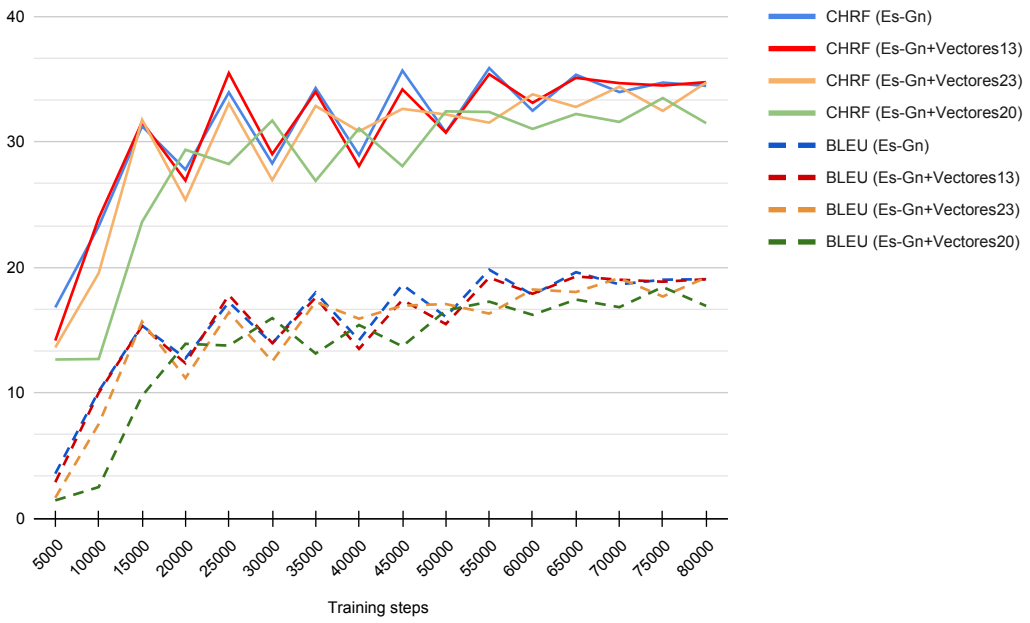


Figura 5.2: Evolución de BLEU y CHRF para los modelos español-guaraní evaluados sobre el conjunto Dev2020-2021.

5.3.1. Resultados

El sentido español-guaraní fue entrenado, en parte, para poder compararnos con los resultados reportados por los participantes de la *shared task* de AmericasNLP en el *Track 2*[1]. En la tabla 5.4 se muestran los resultados obtenidos para el *track* mencionado junto a los obtenidos por nuestros modelos. Todos los modelos se encuentran ordenados en forma de *ranking* de acuerdo a la métrica ChrF obtenida sobre el conjunto TestANLP.

Rank	**Modelo**	**BLEU**	**ChrF**
1	Helsinki-5	6,13	33,6
2	Helsinki-4	4,10	27,6
3	NRC-CNRC-1	2,86	26,1
4	UTokyo-3	3,16	25,4
5	UTokyo-4	2,97	25,1
6	Tamalli-5	1,90	20,7
7	Baseline-1	0,12	19,3
8	Tamalli-3	1,03	18,7
9	Tamalli-1	0,05	17,2
10	**Es-Gn+Vectores23**	**0,17**	**13,0**
11	**Es-Gn**	**0,49**	**12,9**
12	CoAStaL-2	0,03	12,8
13	**Es-Gn+Vectores13**	**0,45**	**12,7**
14	**Es-Gn+Vectores20**	**0,12**	**12,1**
15	Tamalli-2	0,13	10,8

Tabla 5.4: Resultados del *Track 2* de Americas NLP y de los modelos entrenados en este proyecto (resaltados en negrita), rankeados según ChrF.

Si bien nuestros modelos no se encuentran en el último lugar, quedamos muy lejos de la parte superior de la tabla. Al realizar una lectura de los papers de los participantes, comprobamos que realizaron un gran preprocesamiento de los datos, además de utilizar técnicas de aumentación de texto, ponderación de datos en el entrenamiento, modelos multilingües, entre otras. El modelo del equipo ganador, Helsinki, fue entrenado durante más de 200K iteraciones (Vázquez et al. 2021) mientras que los nuestros solamente llegaron a 80K.

[1]En AmericasNLP se realizaron dos *tracks*. En el primero se permitía el uso del conjunto de *development* para el entrenamiento y en el segundo no.

De todos modos, si comparamos el desempeño de nuestros modelos en el resto de conjuntos de *test*, vemos que se comportan bastante similar a los de la otra dirección. Estos resultados se muestran en la tabla 5.5.

Modelo	Step	Test2020		Test2021		DevANLP		TestANLP	
		BLEU	ChrF	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF
`Es-Gn`	55.000	20,55	36,52	20,59	**37,08**	0,27	12,77	0,49	12,91
`Es-Gn+Vectores13`	25.000	20,19	**36,95**	17,33	35,42	0,32	**13,10**	0,45	12,72
`Es-Gn+Vectores23`	80.000	19,75	35,13	20,24	36,23	0,36	12,49	0,17	**13,00**
`Es-Gn+Vectores20`	75.000	18,44	33,74	19,81	35,98	0,23	11,98	0,12	12,06

Tabla 5.5: Experimentos de traducción desde español a guaraní, evaluados sobre Test2020, Test2021, DevANLP y TestANLP.

5.4. Back-Translation

Este último experimento busca evaluar mediante la observación cuánto se distorsiona la oración en español original al traducirla al guaraní y luego volver a traducirla al español. Es importante destacar que esta no es una evaluación formal de los modelos, sino un simple experimento exploratorio para ver el comportamiento de la composición de los mejores traductores construidos.

Para ello elegimos el modelo de cada sentido cuya medida ChrF fue la mayor al evaluarse contra Test2020, por lo que tradujimos las oraciones originales al guaraní utilizando el modelo `Es-Gn+Vectores13` y las volvimos a traducir al español utilizando `Gn-Es+Vectores13`. En la tabla 5.6 mostramos algunas oraciones resultantes de este experimento. Al igual que en la sección anterior cuando mostramos ejemplos de traducción, esta selección no es un análisis exhaustivo del desempeño general de los traductores sino que fueron elegidos por parecernos ejemplos interesantes de comentar.

#	**Original** Tomadas de Test2020	**Back-translation** `Es-Gn+Vectores13` ⟶ `Gn-Es+Vectores13`
1	La ADEC premia a empresas destacadas	Casos de bebés, controlados
2	La Asociación de Empresarios Cristianos (ADEC) premia esta noche a 11 pequeñas y grandes empresas que se diferencian por su aporte social.	La Asociación de Empresarios de Operaciones (CAH) se registró esta noche en el marco de un aporte social.
3	Pasarán año nuevo aislados	Cambian a mediados comunal
4	Durante su discurso inaugural de la premiación, el titular de la ADEC, Juan Luis Ferreira, ahondó sobre la importancia de mejorar el Parlamento, la equidad tributaria, entre otros temas.	El discurso inaugural se desarrolló, el titular de la ADEC, Juan Luis Ferreira, diseñado por el maestro, la equidad tributaria, entre otros temas .
5	Aproximadamente 200 familias (1.000 personas), que abarcan un radio de cinco manzanas, recibieron agua en sus casas de manera sorpresiva esta mañana, luego de que se desmoronara un muro de contención en el Puerto Sajonia.	Casi 200 familias (1.000 personas), a través de un recorrido de 5 páginas, fueron beneficiados con una caja de viaje y a través de la mañana.
6	Su padre fue Juan, señor de Lonvy.	Su padre fue Juan, era de farras del error.
7	Los diputados por Capital	Diputados por Capital
8	Toda la mañana se realizaron mudanzas.	Toda esta mañana.
9	Sin dictamen, Senado tratará paridad democrática	Pronto, el Senado pretende de la paridad
10	La ceremonia se está llevando a cabo en las instalaciones del Banco Central del Paraguay (BCP).	La ceremonia duró aproximadamente las instalaciones del Banco Central del Paraguay (BCP).

Tabla 5.6: Algunos ejemplos del experimento de back-translation.

En primer lugar, contrario a lo que esperábamos, todas las *back-translations* son sorprendentemente fluidas. Por otra parte, como era de esperar, en muchos casos existe tanto ruido que termina provocándose la pérdida total del significado original. Este es el caso de #1, donde la oración original mencionaba un evento de premiación pero la doble traducción genera una salida que habla de bebés. Además, las palabras generadas no conforman una oración o frase sintácticamente correcta. Sucede lo mismo con el ejemplo #3.

En otros casos se logra mantener algunos rasgos de la oración original. Por ejemplo en el #2 vemos que se pierden detalles importantes de la oración original, pero se mantiene que el sujeto es una asociación y el concepto de "aporte social". Sucede lo mismo en el #5, donde se conserva el número de familias y

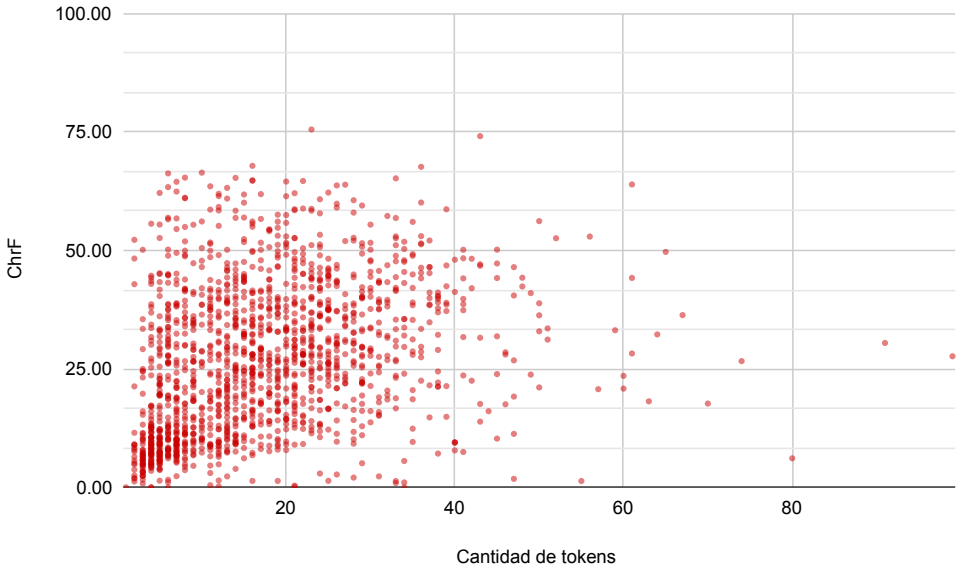
personas, en el #6, en el #8 y en el #10 donde se logran conservar los nombres de entidades pero no la descripción del evento.

Nos vimos gratamente sorprendidos con los resultados de #4, #7 y #9. En #4 se mantiene la situación descrita en la oración original sobre el discurso inaugural del titular de la ADEC. Los ejemplos #7 y #9 son más cortos pero al fijarnos en su significado vemos que apenas pierden detalles.

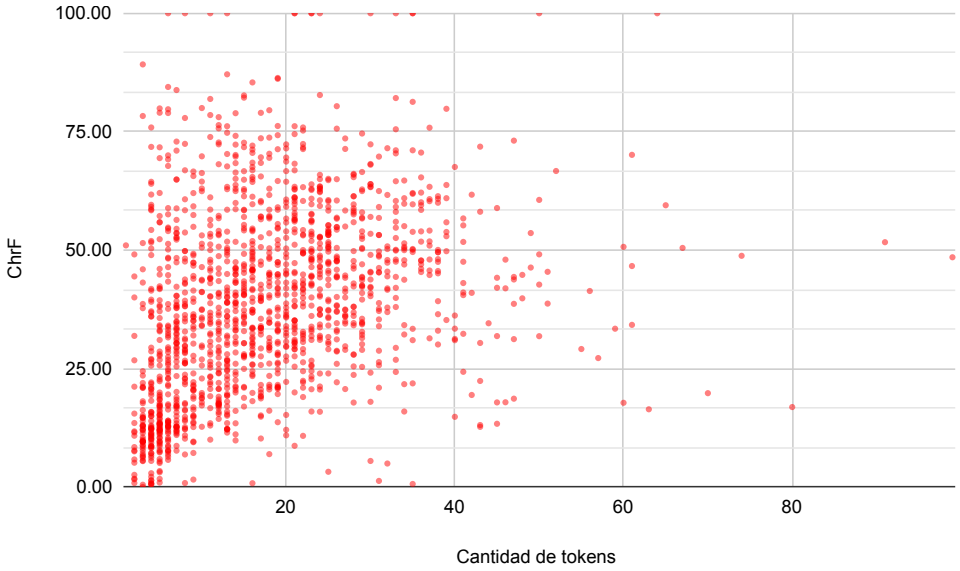
5.5. Relación entre la calidad de la traducción y el largo de la oración

Luego de finalizados los experimentos y observando las salidas obtenidas se nos presentó la duda de si el largo de la oración a traducir impactaba de alguna manera en el desempeño de los modelos construidos. Para esto tomamos los mejores modelos según la medida ChrF sobre Test2020, `Es-Gn+Vectores13` y `Gn-Es+Vectores13`, y realizamos un gráfico de puntos, mostrados en las figuras 5.3a y 5.3b correspondientemente. La opacidad de cada punto es baja, por lo que un incremento en su intensidad significa que hay dos o más puntos en esa posición; es decir, dos o más oraciones con esa cantidad de *tokens* y puntaje ChrF. Es importante destacar que en este caso medimos el ChrF sobre cada oración, en lugar de medirla sobre todo el conjunto a la vez como hicimos para los resultados reportados en la secciones 5.2.1 y 5.3.1.

En ambos casos podemos observar que, a medida que aumenta el largo de las oraciones, los valores bajos de ChrF desaparecen. Quizás esto pueda indicar que las frases cortas son más difíciles de traducir que las demás, o también que las frases largas presentan estructuras gramaticales más predecibles como sujeto-predicado que suelen tener nombres propios, que vimos que logran traducirse mejor en general.



(a) Es-Gn+Vectores13



(b) Gn-Es+Vectores13

Figura 5.3: Valores de ChrF en función de la cantidad de *tokens* de cada oración, obtenidos evaluando sobre el conjunto de Test2020.

Por otra parte, en la traducción hacia el español hay más oraciones con valores de ChrF altos que en el otro sentido, esto puede deberse a que los modelos en la dirección guaraní-español presentaban una *performance* un poco mejor que en el otro sentido. Se observan algunas oraciones que tienen un puntaje de 100 en el tope de la gráfica, que parecen ser *outliers* del modelo. Luego de chequear los datos correspondientes a estos casos atípicos, notamos que hay algunas oraciones que están presentes tanto en el conjunto de *training* como

de *test*. Esto se debe a que en el corpus original (Chiruzzo et al. 2020) hay documentos con mismo contenido pero con distinta fecha, lo cual parecería ser una repetición de noticias por parte de los periódicos.

Sin embargo, los pares de oraciones en esta situación son muy pocas: solamente 10 de ellos están a la vez en el conjunto de *training* y *test*. Consideramos que esta cantidad tan baja no afecta los resultados reportados durante el proyecto.

Para realizar una mejor lectura de estos fenómenos y encontrar fortalezas y debilidades de los modelos, es necesario un análisis más profundo.

Capítulo 6

Conclusiones

En este capítulo presentaremos las conclusiones que surgen de nuestro trabajo. En primer lugar evaluaremos los resultados obtenidos con respecto a los objetivos planteados. Luego detallaremos los recursos generados y listaremos algunos puntos que consideramos pertinente abordar en el trabajo futuro.

6.1. Conclusiones del proyecto

Al comenzar el proyecto nos propusimos recolectar tanto texto paralelo como el presentado en Chiruzzo et al. 2020 y tantos *tweets* como fuera posible. Consideramos que ambos objetivos fueron cumplidos con creces. El conjunto paralelo cuenta con 15.175 pares de oraciones, mientras que la versión del 2020 tiene 14.500. Por su parte, el conjunto de *tweets* cuenta con un total de 9.365 *tweets confiables* (categorías A y B) que suman 111.471 *tokens*.

Los vectores de palabras demostraron capturar información del texto utilizado. Los resultados presentados en Góngora et al. 2021 ya eran muy prometedores, pero las colecciones construidas posteriormente en el proyecto resultaron aún mejores para todas las métricas. Creemos que la falta de diversidad en los textos usados para el entrenamiento genera la gran disparidad mostrada entre los resultados de *family* y *ccc*. Por último, el puntaje obtenido en el test MC-30 da la pauta de que hay cierta correlación entre la similitud calculada y la esperada. En general los resultados de los tests intrínsecos parecen confirmar que la categoría C de *tweets* es muy ruidosa: usándolos no se consiguió ningún

puntaje máximo mientras que sí son usados por modelos que tienen puntajes mínimos. Por último, los resultados para el experimento de visualización de palabras también sugieren que los vectores logran generalizar en cierta medida la información, ya que las palabras de las mismas categorías semánticas tienden a aparecer cerca en el espacio.

En cuanto a la traducción automática, el fin último de este proyecto, logramos buenos resultados. En primer lugar, con el nuevo texto paralelo recolectado se logra una mejora con respecto a los resultados obtenidos por Borges et al. (2021) donde se entrenó solamente sobre el de Chiruzzo et al. 2020. De aquí concluimos que la inclusión de más texto puede ser el principal factor de mejora. Adicionalmente realizamos experimentos en el sentido opuesto en vías de compararnos con los resultados reportados en la *shared task* de AmericasNLP. El resultado obtenido en este caso no fue tan bueno, ya que el texto utilizado para la evaluación en la competencia era de una naturaleza muy distinta del que contábamos para entrenar los modelos. Sin embargo, en cuanto a la medida ChrF sobre Test2020, el desempeño se mantuvo similar a los experimentos en la dirección opuesta. Se notó un incremento en el desempeño al utilizar vectores de palabras que, si bien fue leve, fue consistente entre los diferentes modelos entrenados. En general, la colección de vectores `Vectores13` pareció mejorar los resultados en ambos sentidos, lo cual está en línea con el resultado del test MC-30 en el que dicha colección resultó primera.

6.2. Recursos disponibles

Como se indicó en el documento, a lo largo del proyecto generamos varios recursos que pueden ser de utilidad para futuros trabajos. Estos recursos están disponibles públicamente en GitHub[1] y se detallan a continuación:

- El **artículo** publicado en el *Primer workshop de PLN para Lenguas Indígenas de las Américas*, disponible en la Antología de la ACL[2]. En él se describe la versión inicial del corpus presentado en el capítulo 3 y algunos de los experimentos monolingües descritos en el capítulo 4.

[1] https://github.com/sgongora27/giossa-gongora-guarani-2021
[2] https://aclanthology.org/2021.americasnlp-1.16/ - Accedido por última vez el 22.08.2021.

- **Lista curada de palabras frecuentes**, utilizada para implementar el detector de guaraní detallado en la sección 3.2.2.2.
- **Conjunto paralelo de noticias** descrito en la tabla 3.2, ya alineado a nivel de oración.
  - `parallel_march.zip` contiene el conjunto paralelo reportado en el artículo ya mencionado (Góngora et al. 2021).
  - `parallel_april.zip` contiene el conjunto paralelo usado para los experimentos de este proyecto.
- **Conjuntos de Tweets** descritos en la tabla 3.5, en formato .csv con una columna que indica la categoría del *tweet*.
  - `tweets_gn_march.csv` es el conjunto de *tweets* reportado en el artículo ya mencionado (Góngora et al. 2021).
  - `tweets_gn_june.csv` es el conjunto de *tweets* usado para los experimentos de este proyecto.
  - `tweets_gn_august.csv` es el conjunto final de tweets.
- **Tests traducidos**
  - `capital-common-countries_gn.txt`, traducción estricta del test de analogías original.
  - `family_gn.txt`, inspirado y adaptado del test de analogías original.
  - `MC30_gn.csv`, traducción estricta del test de similitud original, con la colaboración del hablante nativo Marvin Agüero-Torales.
  - `etcheverry_et_al_2016_words_gn.csv`, traducción de las palabras utilizadas en el experimento de clusterización de Etcheverry y Wonsever (2016) con sus categorías, tal como se describe en la tabla 4.3.
- Algunas de las **colecciones de vectores** descritas en el capítulo 4 y evaluadas según diferentes tests.
  - `Vectores09`, de dimensión 150, que no logró de los mejores resultados en el test MC-30 pero obtuvo muy buenos valores en los tests de analogías.
  - `Vectores13`, de dimensión 300, que resultó primera según el test de similitud MC-30.
  - `Vectores19`, de dimensión 300, que obtuvo dos resultados máximos en los tests de analogías.

- Dos de los **modelos de traducción** entrenados: uno en el sentido guaraní-español (`Gn-Es+Vectores13.pt`) y otro en el sentido opuesto (`Es-Gn+Vectores13.pt`). Estos modelos son los que mejor se desempeñaron según la métrica ChrF sobre el corpus Test2020, como se desarrolla en el capítulo 5.

Además publicamos una *demo* en Google Colab[1] que permite probar el modelo `Gn-Es+Vectores13` de traducción guaraní-español, el modelo `Es-Gn+Vectores13` de traducción español-guaraní, el experimento de *back-translation* y el de visualización de palabras.

6.3. Trabajo futuro

Detallamos ahora algunas líneas de trabajo a futuro que podrían seguirse tanto hacia el objetivo de construir un traductor guaraní-español robusto como para el trabajo monolingüe sobre el guaraní.

6.3.1. Sobre el corpus

Acorde a los resultados publicados en el artículo aceptado en *AmericasNLP* (Góngora et al. 2021) y el análisis de resultados presentado aquí, la mayor barrera impuesta por el corpus es la de la temática, como se detalló en la sección 4.2.4 . Hay muchas palabras que no están en el corpus o están pero con una frecuencia ínfima. Se necesitan más textos enciclopédicos y literarios, donde se aborden conceptos que no se mencionan en los textos periodísticos. Un posible enfoque es automatizar la periódica descarga y limpieza del último *dump* de la Wikipedia, de modo de siempre contar con la última versión disponible para los entrenamientos. Si bien la Wikipedia en guaraní continúa siendo un recurso mucho más pequeño que la correspondiente a otras lenguas, crece día a día gracias a sus colaboradores activos[2].

Por otra parte sería de mucha utilidad hacer un curado manual de la categoría C de los tweets. Tal como se explicó en la sección 3.2.3, esta categoría

[1] https://drive.google.com/drive/folders/1cZP67qwImo2pXcaACoPoNp1c0yhWEzoE

[2] https://gn.wikipedia.org/wiki/Mba%27ech%C4%A9ch%C4%A9:Estad%C3%ADsticas - Accedido por última vez el 22.08.2021.

contiene *tweets* con cantidad dispar de guaraní. Hay *tweets* que están prácticamente en español, algunos pocos que están en otras lenguas y otros que tienen alguna expresión o están completamente en guaraní. Muchos de los 87.633 *tweets* de la categoría C pueden aprovecharse para ampliar el conjunto de 9.365 *tweets confiables* de la versión final. De igual manera consideramos prioritario curar la alineación del conjunto paralelo de noticias, tal como se hizo para el construido por Chiruzzo et al. (2020) con hablantes nativos.

Algo que se puede intentar es repetir el crawling de webs detallado en la sección 3.1.1 pero con la lista curada utilizada para la construcción del conjunto de tweets. Si bien los sitios web en general y las redes sociales son dominios totalmente diferentes, los problemas reportados durante el crawling son los mismos que intenta resolver el método de lista de palabras detallado en la sección 3.2.2.2.

Mirando hacia el futuro, consideramos importante que se continúe recolectando texto de redes sociales en guaraní, pues reflejan el uso actual de la lengua, tanto a nivel léxico como a nivel temático. Para esto vemos útil la construcción de un identificador de lengua **estadístico** que permita determinar si los *tweets* están en guaraní/jopará o no. Para esto debería entrenarse sobre texto en otras lenguas presentes en la categoría C de nuestro conjunto, como el portugués, español o tailandés. Un identificador de esta naturaleza permitiría filtrar los *tweets* sin apegarse estrictamente a una lista de palabras, lo que en este proyecto implicó que no pudiéramos desambiguar los *tweets* de la categoría C por contener palabras que no estaban entre las frecuentes (como se explicó en la sección 3.2.3).

6.3.2. Sobre los vectores de palabras y los tests intrínsecos

Es necesario contar con más variedad de tests de manera de poder medir la calidad de los vectores con diferentes conjuntos de palabras y con mayor diversidad léxica. De los conjuntos disponibles para tests de similitud solamente tradujimos MC-30, que es conocido por ser el más chico y con palabras menos particulares, por lo que podrían traducirse otros como SimLex-999 o WordSimilarity-353. Sin embargo, cuanto menos frecuentes sean las palabras

presentes en los tests, mayor es la cantidad de texto con que se debe contar para entrenar los vectores a evaluar, como mencionábamos anteriormente.

También consideramos muy importante probar con otros métodos para generar los vectores. En primer lugar, evaluar y comparar el desempeño al usar el algoritmo *skip-gram*, en lugar de *c-bow*, a la hora de construir los vectores con *word2vec*. En segundo lugar entrenar vectores usando *fastext* que, al tomar en cuenta n-gramas a nivel de caracter, puede resultar mejor para lenguajes morfológicamente ricos (Bojanowski et al. 2017) como el guaraní. Una evaluación y comparación robusta podría dar luz sobre cuál es el algoritmo o método más adecuado para cada caso.

Al día de la fecha *Polyglot* no dispone de vectores de palabras para el guaraní[1]. Es importante comprobar periódicamente esta condición, pues de publicarse pueden resultar de gran utilidad tanto para realizar comparaciones como para usarlos en otras tareas.

Por último, sería bueno probar su *performance* en otras tareas más allá de traducción automática, como análisis de sentimientos o *question-answering*.

6.3.3. Sobre la traducción automática

La limitante de estar atados a las restricciones de uso de *Google Colab* no nos permitió hacer más experimentos que los mostrados. Sin ir más lejos, solamente pudimos probar 3 de los 24 modelos de vectores de palabras entrenados para el guaraní. Por otra parte, creemos importante ahondar en la configuración de OpenNMT, como diferentes arquitecturas neuronales, técnicas de regularización para evitar caer en *overfitting* o ponderación de los corpus de entrenamiento.

En cuanto a los resultados obtenidos, aún queda muchísimo margen de mejora en ambas direcciones (español a guaraní y guaraní a español). Al igual que comentamos anteriormente, se necesita texto más diverso para que la traducción sea más robusta. Esto puede verse al comparar los resultados de la

[1] https://polyglot.readthedocs.io/en/latest/Download.html - Accedido por última vez el 22.08.2021.

evaluación sobre el texto paralelo de noticias y el conjunto de AmericasNLP, que está conformado en gran parte por diálogos. Sin embargo es importante resaltar que, ante la escasez de recursos, cualquier texto paralelo que pueda recolectarse será útil pues vimos que impacta de gran manera en la *performance* de los traductores. Está claro que futuros modelos de traducción pueden ser entrenados utilizando este conjunto traducido para la *shared task* de AmericasNLP.

De forma de mejorar los resultados obtenidos, próximos trabajos podrían centrarse en un mejor filtrado y preprocesamiento de los recursos utilizados, además de explorar técnicas de aumentación de texto mediante *back-translation*, lo cual parece ser útil en contextos de escasez de recursos (Abdulmumin et al. 2021; Caswell et al. 2019; Edunov et al. 2018).

Bibliografía

Abdelali, A., Cowie, J., Helmreich, S., Jin, W., Milagros, M. P., Ogden, B., Rad, H. M. y Zacharski, R. (2006). Guarani: a case study in resource development for quick ramp-up MT. *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas,"Visions for the Future of Machine Translation*, 1-9.

Abdulmumin, I., Galadanci, B. S. y Isa, A. (2021). Enhanced Back-Translation for Low Resource Neural Machine Translation Using Self-training. En S. Misra y B. Muhammad-Bello (Eds.), *Information and Communication Technology and Applications* (pp. 355-371). Springer International Publishing.

Academia de la Lengua Guaraní. (2018). *Gramática guaraní.* ".Editorial Servilibro".

Agüero-Torales, M., Vilares, D. y López-Herrera, A. (2021). On the logistical difficulties and findings of Jopara Sentiment Analysis. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, 95-102. https://doi.org/10.18653/v1/2021.calcs-1.12

Alabdulkarim, A., Li, S. y Peng, X. (2021). Automatic Story Generation: Challenges and Attempts, 72-83. https://doi.org/10.18653/v1/2021.nuse-1.8

Alcaraz, N. A. y Alcaraz, P. A. (2020). Aplicación web de Análisis y Traducción Automática Guaraní–Español/Español–Guaraní. *Revista Científica de la UCSA*, *7*(2), 41-69.

Al-Rfou', R., Perozzi, B. y Skiena, S. (2013). Polyglot: Distributed Word Representations for Multilingual NLP, 183-192. https://aclanthology.org/W13-3520

Azzinnari, A. y Martínez, A. (2016). *Representación de Palabras en Espacios de Vectores* (Proyecto de grado). Universidad de la República. Uruguay.

Bahdanau, D., Cho, K. y Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bird, S., Klein, E. y Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. ".O'Reilly Media, Inc.".

Bojanowski, P., Grave, E., Joulin, A. y Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.

Bollmann, M., Aralikatte, R., Murrieta Bello, H., Hershcovich, D., de Lhoneux, M. y Søgaard, A. (2021). Moses and the Character-Based Random Babbling Baseline: CoAStaL at AmericasNLP 2021 Shared Task, 248-254. https://doi.org/10.18653/v1/2021.americasnlp-1.28

Borges, Y. y Mercant, F. (2019). *Herramientas para traduccion automatica Guarani - Español*. Online, Universidad de la Republica (Uruguay). https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/24378

Borges, Y., Mercant, F. y Chiruzzo, L. (2021). Using Guarani Verbal Morphology on Guarani-Spanish Machine Translation Experiments. *Procesamiento del Lenguaje Natural*, *66*, 89-98.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L. y Roossin, P. S. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, *16*(2), 79-85. https://aclanthology.org/J90-2002

Çano, E. y Morisio, M. (2019). Word Embeddings for Sentiment Analysis: A Comprehensive Empirical Survey. *ArXiv*, *abs/1902.00753*. https://arxiv.org/pdf/1902.00753.pdf

Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J. y Way, A. (2017). Is Neural Machine Translation the New State of the Art? *The Prague Bulletin of Mathematical Linguistics*, *108*, 109-120. https://doi.org/10.1515/pralin-2017-0013

Caswell, I., Chelba, C. y Grangier, D. (2019). Tagged Back-Translation. *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, 53-63. https://doi.org/10.18653/v1/W19-5206

Chen, B. y Cherry, C. (2014). A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU. *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 362-367. https://doi.org/10.3115/v1/W14-3346

Chiruzzo, L., Amarilla, P., Ríos, A. y Giménez Lugo, G. (2020). Development of a Guarani - Spanish Parallel Corpus, 2629-2633. https://aclanthology.org/2020.lrec-1.320

Cho, K., van Merrienboer, B., Bahdanau, D. y Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR, abs/1409.1259*. http://arxiv.org/abs/1409.1259

Colton, S. y Wiggins, G. (2012). Computational creativity: The final frontier? *Frontiers in Artificial Intelligence and Applications*, *242*, 21-26. https://doi.org/10.3233/978-1-61499-098-7-21

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S. R., Schwenk, H. y Stoyanov, V. (2018). XNLI: Evaluating Cross-lingual Sentence Representations. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Devlin, J., Chang, M.-W., Lee, K. y Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Dooley, R. A. (2006). Léxico guarani, dialeto mbyá com informações úteis para o ensino médio, a aprendizagem e a pesquisa lingüística. *Cuiabá, MT: Sociedade Internacional de Lingüística*, *143*, 206.

Edunov, S., Ott, M., Auli, M. y Grangier, D. (2018). Understanding Back-Translation at Scale. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 489-500. https://doi.org/10.18653/v1/D18-1045

Estigarribia, B. y Pinta, J. (2017). *Guarani Linguistics in the 21st Century*. Brill. https://doi.org/https://doi.org/10.1163/9789004322578

Etcheverry, M. y Wonsever, D. (2016). Spanish Word Vectors from Wikipedia, 3681-3685. https://www.aclweb.org/anthology/L16-1584

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. y Ruppin, E. (2002). Placing Search in Context: The Concept Revisited. *ACM Trans. Inf. Syst.*, *20*(1), 116-131. https://doi.org/10.1145/503104.503110

García-Vega, M., Díaz-Galiano, M., García-Cumbreras, M., Plaza-Del-Arco, F., Montejo-Ráez, A., Zafra, S. M., Martínez-Cámara, E., Aguilar, C., Antonio, M., Cabezudo, S., Chiruzzo, L. y Moctezuma, D. (2020). Overview of TASS 2020: Introducing Emotion Detection. http://ceur-ws.org/Vol-2664/tass_overview.pdf

Garg, A. y Agarwal, M. (2019). Machine Translation: A Literature Review. *CoRR, abs/1901.01122*. http://arxiv.org/abs/1901.01122

Gasser, M. (2018). Mainumby: un Ayudante para la Traducción Castellano-Guaraní. *arXiv preprint arXiv:1810.08603*.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H. M., III, H. D. y Crawford, K. (2018). Datasheets for Datasets. *CoRR, abs/1803.09010*. http://arxiv.org/abs/1803.09010

Gervás, P., Concepción, E., León, C., Méndez, G. y Delatorre, P. (2019). The long path to narrative generation. *IBM Journal of Research and Development, 63*(1), 8:1-8:10. https://doi.org/10.1147/JRD.2019.2896157

Góngora, S., Giossa, N. y Chiruzzo, L. (2021). Experiments on a Guarani Corpus of News and Social Media, 153-158. https://doi.org/10.18653/v1/2021.americasnlp-1.16

Hill, F., Reichart, R. y Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics, 41*(4), 665-695. https://doi.org/10.1162/COLI_a_00237

Jauhiainen, H., Jauhiainen, T. y Lindén, K. (2020). Building Web Corpora for Minority Languages. *Proceedings of the 12th Web as Corpus Workshop*, 23-32. https://www.aclweb.org/anthology/2020.wac-1.4

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T. y Lindén, K. (2018). Automatic Language Identification in Texts: A Survey. *CoRR, abs/1804.08186*. http://arxiv.org/abs/1804.08186

Joshi, P., Santy, S., Budhiraja, A., Bali, K. y Choudhury, M. (2020). The State and Fate of Linguistic Diversity and Inclusion in the NLP World. https://doi.org/10.18653/v1/2020.acl-main.560

Jurafsky, D. y Martin, J. H. (2009). *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc.

Jurafsky, D. y Martin, J. H. (2021). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd - Draft). https://web.stanford.edu/~jurafsky/slp3/

Kalchbrenner, N. y Blunsom, P. (2013). Recurrent Continuous Translation Models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1700-1709. https://aclanthology.org/D13-1176

Klein, G., Kim, Y., Deng, Y., Senellart, J. y Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *CoRR, abs/1701.02810*. http://arxiv.org/abs/1701.02810

Knowles, R., Stewart, D., Larkin, S. y Littell, P. (2021). NRC-CNRC Machine Translation Systems for the 2021 AmericasNLP Shared Task. *Procee-*

dings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, 224-233. https://doi.org/10.18653/v1/2021.americasnlp-1.25

Kunc, L. y Saravia, M. (2020). *Identificación de discurso de odio en redes sociales.* Online, Universidad de la Republica (Uruguay). https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/25263

Kusner, M., Sun, Y., Kolkin, N. y Weinberger, K. (2015). From word embeddings to document distances. *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 957-966. http://proceedings.mlr.press/v37/kusnerb15.pdf

Kuznetsova, A. y Tyers, F. (2021). A finite-state morphological analyser for Paraguayan Guaraní. *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, 81-89. https://doi.org/10.18653/v1/2021.americasnlp-1.9

Lenci, A. (2018). Distributional Models of Word Meaning.

Luong, M.-T., Pham, H. y Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *CoRR, abs/1508.04025.* http://arxiv.org/abs/1508.04025

Lustig, W. (2010). Mba'éichapa oiko la guarani? Guaraní y jopara en el Paraguay. *PAPIA-Revista Brasileira de Estudos do Contato Linguístico*, *4*(2), 19-43.

Maćkiewicz, A. y Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, *19*(3), 303-342. https://doi.org/https://doi.org/10.1016/0098-3004(93)90090-R

Mager, M., Oncevay, A., Ebrahimi, A., Ortega, J., Rios, A., Fan, A., Gutierrez-Vasques, X., Chiruzzo, L., Giménez-Lugo, G., Ramos, R., Meza Ruiz, I. V., Coto-Solano, R., Palmer, A., Mager-Hois, E., Chaudhary, V., Neubig, G., Vu, N. T. y Kann, K. (2021). Findings of the AmericasNLP 2021 Shared Task on Open Machine Translation for Indigenous Languages of the Americas, 202-217. https://doi.org/10.18653/v1/2021.americasnlp-1.23

Michael, L., Chousou-Polydouri, N., Bartolomei, K., Donnelly, E., Meira, S., Wauters, V. y O'Hagan, Z. (2015). A Bayesian Phylogenetic Classification of Tupí-Guaraní. *LIAMES: Línguas Indígenas Americanas*, *15*(2), 193-221. https://doi.org/10.20396/liames.v15i2.8642301

Mikolov, T., Chen, K., Corrado, G. y Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space.

Mikolov, T., Yih, W.-t. y Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations, 746-751. https://www.aclweb.org/anthology/N13-1090

Miller, G. A. y Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*(1), 1-28. https://doi.org/10.1080/01690969108406936

Minnen, G., Carroll, J. y Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, *7*, 207-223.

Mitchell, T. M. (1997). *Machine Learning* (1.ª ed.). McGraw-Hill, Inc.

Nagoudi, E. M. B., Chen, W.-R., Abdul-Mageed, M. y Cavusoglu, H. (2021). IndT5: A Text-to-Text Transformer for 10 Indigenous Languages. *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, 265-271. https://doi.org/10.18653/v1/2021.americasnlp-1.30

Papineni, K., Roukos, S., Ward, T. y Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation, 311-318. https://doi.org/10.3115/1073083.1073135

Parida, S., Panda, S., Dash, A., Villatoro-Tello, E., Doğruöz, A. S., Ortega-Mendoza, R. M., Hernández, A., Sharma, Y. y Motlicek, P. (2021). Open Machine Translation for Low Resource South American Languages (AmericasNLP 2021 Shared Task Contribution), 218-223. https://doi.org/10.18653/v1/2021.americasnlp-1.24

Pennington, J., Socher, R. y Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543. http://www.aclweb.org/anthology/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. y Zettlemoyer, L. (2018). Deep contextualized word representations. *Proc. of NAACL*.

Popović, M. (2015). chrF: character n-gram F-score for automatic MT evaluation. *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392-395. https://doi.org/10.18653/v1/W15-3049

Popović, M. (2017). chrF++: words helping character n-grams. *Proceedings of the Second Conference on Machine Translation*, 612-618. https://doi.org/10.18653/v1/W17-4770

Qi, Y., Sachan, D., Felix, M., Padmanabhan, S. y Neubig, G. (2018). When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 529-535. https://doi.org/10.18653/v1/N18-2084

Řehůřek, R. y Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora, 45-50.

Ríos, A. A., Amarilla, P. J. y Lugo, G. A. G. (2014). Sentiment categorization on a creole language with lexicon-based and machine learning techniques. *2014 Brazilian Conference on Intelligent Systems*, 37-43. https://ieeexplore.ieee.org/document/6984804

Rogers, A., Kovaleva, O. y Rumshisky, A. (2020). A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, *8*, 842-866. https://doi.org/10.1162/tacl_a_00349

Rosa, J. (2013). Historiografía lingüística del Río de la Plata: las lenguas indígenas de la Banda Oriental. *Boletín de filología*, *48*, 131-171. https://doi.org/10.4067/S0718-93032013000200007

Rudnick, A., Skidmore, T., Samaniego, A. y Gasser, M. (2014). Guampa: a Toolkit for Collaborative Translation. *LREC*, 1659-1663.

Secretaría de Políticas Lingüísticas del Paraguay. (2019). Corpus de Referencia del Guaraní Paraguayo Actual – COREGUAPA.

Sutskever, I., Vinyals, O. y Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *CoRR*, *abs/1409.3215*. http://arxiv.org/abs/1409.3215

Thomas, G. (2019). Universal Dependencies for Mbyá Guaraní. *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, 70-77.

Van de Cruys, T. (2020). Automatic Poetry Generation from Prosaic Text, 2471-2480. https://doi.org/10.18653/v1/2020.acl-main.223

Vázquez, R., Scherrer, Y., Virpioja, S. y Tiedemann, J. (2021). The Helsinki submission to the AmericasNLP shared task, 255-264. https://doi.org/10.18653/v1/2021.americasnlp-1.29

Vitale, M. A. (2012). Representaciones del guaraní y el español en alumnado de Escuelas medias de la ciudad de Buenos Aires con población de familias

guaraní hablantes. *Lingüística*, *28*, 21-40. http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S2079-312X2012000100003&nrm=iso

Wang, B., Wang, A., Chen, F., Wang, Y. y Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, *8*, e19. https://doi.org/10.1017/ATSIP.2019.12

Zarratea, T. (2009). El guaraní: la lengua americana más viable. *Revista paraguaya de sociología : publicación de ciencias sociales para América Latina*, *46*, 35-49.

Zheng, F., Reid, M., Marrese-Taylor, E. y Matsuo, Y. (2021). Low-Resource Machine Translation Using Cross-Lingual Language Model Pretraining. *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, 234-240. https://doi.org/10.18653/v1/2021.americasnlp-1.26

Glosario

Presentamos aquí una lista ordenada alfabéticamente de algunos términos usados en el documento.

ACL *Association for Computational Linguistics* (Asociación de lingüística computacional) es la sociedad científica para las personas que trabajan en PLN.

Alineación Es el proceso de definir la correspondencia que hay entre cada par de elementos de un conjunto paralelo. La alineación puede hacerse a nivel de documento, de oración o de palabra. Todos los conjuntos paralelos mencionados en este trabajo están alineados a nivel de oración.

AmericasNLP *Workshop* sobre lenguas indígenas de las américas. Su primera edición tuvo lugar junto a la NAACL2021, donde parte de este trabajo fue publicado.

Capital-common-countries (ccc) Test de analogías de palabras que trata de capitales y países comúnmente nombrados en la gran mayoría de corpus.

Checkpoint Durante el entrenamiento de los modelos de traducción automática se fueron persistiendo versiones cada 5.000 iteraciones. Un checkpoint corresponde a una de esas versiones guardadas.

Code switching Fenómeno lingüístico que consiste en el cambio de lengua durante una conversación o expresión.

Conjunto de development Subconjunto de un corpus destinado exclusivamente a realizar ajustes a un modelo durante su desarrollo.

Conjunto de test Subconjunto de un corpus destinado exclusivamente a ser usado en la evaluación de un modelo.

Conjunto de training Subconjunto de un corpus destinado exclusivamente a ser usado en el entrenamiento de un modelo.

Corpus Colección de textos recopilados con algún objetivo, como por ejemplo reflejar el uso de una lengua en particular.

Corpus anotado Corpus que cuenta con algún tipo de información adjunta, como por ejemplo la lengua en que está escrito.

Corpus paralelo Corpus conformado por varios textos en diferentes lenguas que se corresponden entre sí de alguna manera, ya sea a nivel de documento, de oración o de palabra.

Crawling Proceso de navegar y descargar automáticamente una serie de sitios web mediante el uso de un programa comúnmente llamado *crawler*.

Cron job Conjunto de comandos a ejecutarse periódicamente.

Family Test de analogías de palabras que trata de relaciones familiares y sus variaciones utilizando los géneros masculino y femenino.

Gensim Biblioteca de código abierto para Python que permite la construcción de colecciones de vectores de palabras, entre otras funcionalidades.

Guaraní Lengua perteneciente a la familia *tupí-guaraní* originaria de Sudamérica. En la actualidad es hablado por más de 10 millones de personas.

Jopará Dialecto hablado en Paraguay que surge de la mezcla entre el español y el guaraní.

MC-30 Test de similitud de palabras compuesto por 30 pares de palabras, donde cada uno tiene un puntaje entre 0 y 10 asignado según el juicio humano.

N-grama Secuencia de n elementos consecutivos extraídos de un texto. Estos elementos pueden ser palabras o caracteres.

NAACL *North American Chapter of the Association for Computational Linguistics* (Capítulo Norteamericano de la Asociación de Lingüística computacional) es la organización regional para miembros de la ACL en las américas.

NLTK Biblioteca de código abierto para Python que brinda herramientas básicas para el trabajo en PLN.

OpenNMT *Framework* de código abierto que brinda herramientas para trabajar con traducción automática basada en redes neuronales.

PCA Técnica estadística que, en nuestro proyecto, nos permitió reducir las dimensiones de las colecciones de vectores para realizar el experimento de visualización de palabras.

PLN El *procesamiento de lenguaje natural* es la disciplina de las ciencias de la computación que se enfoca en problemas que involucran el procesamiento de las lenguas habladas por los humanos.

Preprocesamiento de texto Proceso presente en casi todos los sistemas de PLN que consiste en llevar el texto a un formato que facilite su procesamiento. Los detalles de este proceso dependen del problema a resolver.

Query Consulta que se realiza a una API o base de datos. En nuestro trabajo realizamos *queries* a la API de Twitter para obtener *tweets* en guaraní.

Scrapy *Framework* de código abierto implementado en Python que permite la construcción de *web crawlers*, entre otras funcionalidades.

Test de analogías Test intrínseco que busca evaluar la capacidad del modelo de deducir analogías entre palabras tal como lo hacemos los humanos.

Test de similitud Test intrínseco que busca evaluar la capacidad del modelo de evaluar el parecido entre dos conceptos del mundo tal como lo hacemos los humanos.

Test extrínseco Una de las maneras de evaluar colecciones de vectores de palabras. Busca analizar cómo la colección logra reforzar el desempeño en una tarea particular, como por ejemplo la traducción automática.

Test intrínseco Una de las maneras de evaluar colecciones de vectores de palabras. Busca centrarse en las relaciones entre las representaciones de las palabras, independientemente de una tarea en particular.

Tokenizar Separar un texto en *tokens*. Los *tokens* resultantes suelen ser palabras, pero también pueden ser fechas o signos de puntuación. En el contexto de la red social Twitter, un token puede ser también una mención a un usuario ("@Usuario_123"), un hashtag ("#Ejemplo") o un emoji. En algunos contextos también se le llama *tokenización de oración* al proceso de separar un texto en oraciones.

Word embeddings Técnica para modelar las palabras como vectores densos, según el contexto en el que aparecen.