Proyecto de Grado

Ingeniería en Computación

Herramientas para Traducción Automática Guaraní-Español

Yanina Borges Florencia Mercant

Tutor: Luis Chiruzzo

Universidad de la República

Facultad de Ingeniería

Junio 2020
Montevideo, Uruguay

Resumen

Este trabajo se centra en la realización de una herramienta que contribuya a la traducción del guaraní al español. El guaraní es un idioma utilizado actualmente en diferentes países de América Latina, es un idioma aglutinante y polisintético que presenta muchas diferencias respecto al español y al inglés.

El proyecto se divide en tres grandes partes. La primera consiste en la investigación de las herramientas y recursos existentes para la traducción del idioma guaraní al español. Se realizaron estudios acerca del idioma y su gramática, teniendo como eje principal de la investigación los verbos. La segunda parte consiste en elaborar herramientas que faciliten la traducción. Para esto se siguen tres grandes objetivos: realizar el análisis morfológico de verbos, detectar verbos y realizar experimentos de traducción automática guaraní-español. Para realizar el análisis de verbos se implementó una herramienta basada en reglas. La parte detección de verbos se desarrolló en base al método basado en reglas y experimentos basados en un modelo estadístico con Hidden Markov Model. Luego para la traducción automática se realizaron diferentes experimentos con un método basado en redes neuronales, mediante la herramienta OpenNMT e incorporando los conocimientos generados previamente. La última etapa del proyecto consiste en la evaluación de las herramientas implementadas.

Palabras Claves

idioma aglutinante, idioma polisintético, guaraní, jopará, procesamiento de lenguaje natural, análisis morfológico, análisis morfosintáctico, part-of-speech tagging, aprendizaje basado en reglas, openNMT, Hidden Markov Model.

Índice general

Capítulo 1

Introducción

El guaraní es una lengua hablada por 12 millones de personas en varios países de Latinoamérica. Hasta el momento existen pocos trabajos que se hayan enfocado en construir herramientas de análisis para el idioma guaraní, es un idioma poco explorado en cuanto al procesamiento de lenguaje natural, por lo que es una oportunidad de construir herramientas nuevas de gran aplicabilidad.

1.1. Objetivos Generales

El proyecto tiene como objetivo principal contribuir con la investigación del idioma guaraní enfocándonos en el área de Procesamiento del Lenguaje Natural. Este punto ha sido poco estudiado hasta el momento, y se pretende generar un punto de partida para futuros avances en el área. Con el presente proyecto se pretende agregar visibilidad al lenguaje y dar lugar a la investigación del mismo con la incorporación de diferentes tecnologías.

1.2. Objetivos Específicos

El objetivo es lograr una línea base para la traducción automática entre guaraní-español. Para esto se abordaron cuatro puntos específicos:

- Estudio del idioma guaraní, en particular se realizará un estudio exhaustivo de los verbos. Dado que el guaraní es un lenguaje aglutinante y polisintético los verbos son un tipo de unidad gramatical que posee mucha información relevante, pudiendo usarse como base para la comprensión del idioma. Es por esto que gran parte del proyecto se centrará en la detección y análisis morfológico de verbos.

- Creación de recursos computacionales para trabajar con el idioma guaraní. Luego de realizar una recopilación de recursos lingüísticos como diccionarios en línea, corpus paralelos, y libros sobre la gramática del guaraní se construirá un diccionario que será llevado a una base de datos. Esta información estructurada es fácil de acceder y manipular. Es una herramienta que podrá seguir expandiéndose fácilmente a medida que se vayan encontrando o generando nuevos recursos lingüísticos.

- Construcción de una herramienta para detección y análisis de verbos en guaraní. Para el análisis morfológico de palabras se construirá un analizador basado en reglas gramaticales. A partir de este analizador se implementará una herramienta web para consultar los posibles análisis de una palabra. Para la detección de verbos se utilizarán dos enfoques, un modelo basado en reglas gramaticales y un modelo probabilístico basado en Hidden Markov Models. Además se incorporará en la herramienta web desarrollada la posibilidad de realizar consultas que tendrán como entrada una oración y como resultado se obtendrán los verbos encontrados en la misma a partir de una heurística creada en base al método basado en reglas.

- Experimentos de traducción automática guaraní-español para tratar de crear una línea base de traducción del idioma e intentar mejorarla. Se abordará el problema utilizando un método de traducción automática basado en redes neuronales mediante el framework open source OpenNMT. Además de la línea base de traducción proporcionada

por OpenNMT, se realizarán diferentes experimentos para mejorar los resultados.

1.3. Estructura del informe

En esta sección se detalla la estructura del resto del informe.

En el capítulo dos se desarrollan los conceptos necesarios para entender el desarrollo de la investigación realizada. Cuenta con cuatro grandes secciones en las que se detallan la gramática del idioma guaraní, luego una introducción al procesamiento del lenguaje natural, posteriormente se prosigue a explicar algunos conceptos de la traducción automática y finalmente se presenta el relevamiento de recursos lingüísticos encontrados en relación al idioma.

En el capítulo tres se presenta el desarrollo de la investigación. En este se explican algunos aspectos sobre el corpus utilizado y la implementación de las soluciones construidas para los diferentes problemas planteados. También se detallan los recursos tecnológicos utilizados a lo largo del proyecto, se describe la herramienta web construida, su interfaz y las instrucciones básicas para ejecutar sus funcionalidades.

El capítulo cuatro expone los resultados obtenidos para los diferentes problemas planteados y el análisis realizado para los mismos. La evaluación de los resultados fue realizada con medidas aplicables para cada tipo de problema.

En el último capítulo se detallan las conclusiones obtenidas de los resultados y posibles mejoras que se pueden realizar a futuro.

Capítulo 2

Marco teórico

En este capítulo se explicarán conceptos necesarios para lograr comprender ciertos aspectos del proyecto. Se hará una introducción a la gramática del guaraní con énfasis en los verbos. Luego se realizará una revisión del área de PLN, enfocándose en las tareas relevantes para este trabajo. También se explicarán diferentes enfoques con los cuales es posible abordar el problema de traducción automática. Y por último se presentará un relevamiento del estado del arte en cuanto a herramientas y recursos lingüísticos para el guaraní.

2.1. Gramática

El idioma guaraní cuenta con más de 50 variantes indígenas extendidas en América Latina. Paraguay es el país que más utiliza el idioma guaraní, siendo éste el idioma más utilizado según un censo realizado en el año 2002. Se hablan 6 variantes del idioma en diferentes puntos del territorio incluyendo todos los estratos sociales y culturales. En nuestro caso de estudio nos centraremos en el "Jopará" que es una variante no pura del guaraní que se habla en Paraguay. Otras variantes de la familia guaraní son habladas en regiones de Brasil, Bolivia y Argentina.

A diferencia del español donde los accidentes verbales como tiempo, número, persona, voz y modo se expresan como alteraciones de la raíz del verbo, el guaraní representa estos accidentes añadiendo morfemas a la raíz del verbo.

Por ejemplo, el verbo conjugado "aguata" en español se traduce como "camino"(primera persona del singular del verbo caminar). Se puede observar que se forma de la composición del prefijo "a" y la raíz "guata" que significa el verbo "caminar". El prefijo indica la persona que está realizando la acción, en este caso "yo". Mientras que el lema "guata" se traduce al español como el verbo en infinitivo "caminar".

Este ejemplo muestra una de las principales características del idioma guaraní que es su condición aglutinante: sus palabras se forman combinando prefijos y sufijos entorno a un lema o núcleo. Se pueden generar vocablos de varias sílabas formadas por la yuxtaposición de morfemas. Esos morfemas en general se mantienen incambiados al concatenarlos a las palabras y cada uno tiene un significado propio. Se lo conoce también como un lenguaje polisintético, es decir el hablante construye su propia palabra uniendo partículas al lema base con contenido semántico. Las palabras desde un punto de vista morfológico se generan componiendo varios morfemas que son como unidades de significado. En la escritura, las partículas prefijas y sufijas que modifican al lema base, se unen al mismo formando con él una sola palabra.

Extendiendo el ejemplo anterior, consideremos la oración "che aguata ekuélape" que en español significa "yo camino a la escuela". Esta oración tiene como sujeto "che", seguido del verbo conjugado "aguata" explicado

anteriormente, y por el último el complemento “ekuélape”. Este complemento a su vez se puede descomponer en el sustantivo “ekuéla” y el sufijo “pe”, siendo “pe” el equivalente a las palabras “a” o “hacia”, conocidas en español como preposiciones. A diferencia del español donde las preposiciones son palabras en sí mismas, en guaraní se expresan como sufijos del sustantivo.

Por otro lado, se puede observar que la palabra “ekuéla” es un hispanismo, es decir, es una palabra adaptada del español a la ortografía y estructura del guaraní. Esta es una característica propia del Jopará. En guaraní no existe la letra “c” por lo que se reemplaza por la “k”, además se usa el tilde en la “e” para que se acentúe como “escuela”, cosa que en español no es necesario.

El idioma cuenta con 33 letras o grafemas que representan fonemas, no existen las letras mudas como la “h” en el español. Las letras se dividen en 12 vocales, y 21 consonantes. A su vez cada uno de estos grupos se dividen en nasales y orales, esta sub-división se basa en la fonética de las letras. En la tabla 2.1 se muestra la categorización del alfabeto.

	Vocales	Consonantes
Nasales	â - ê - î - ô - û – ŷ	ĝ - m - n - ñ - mb - nd - ng- nt
Orales	a - e - i - o - u – y	ch - g - h - j - k - l - p - r - rr - s - t - v - (’)

Cuadro 2.1: Alfabeto Guaraní

Una letra particular del alfabeto es la consonante glotal puso (’), que solo se utiliza entre vocales. Esta forma sílaba con la vocal que le sigue. El uso del puso crea diferencias semánticas, por ejemplo, “kua” significa agujero, sin embargo “ku’a” significa cintura.

En cuanto a los acentos, al igual que en el español se utilizan para dar énfasis a la vocal tónica o de mayor intensidad, por ejemplo, en la palabra “ekuéla” se utiliza para dar énfasis a la letra “e”. Como regla general nunca se encontrará el acento en la última letra de una palabra, ni en una vocal nasal. Si en una palabra existen dos o más vocales tónicas, el tilde se marca sobre la tónica de la derecha, siempre que no sea vocal final.

En el idioma guaraní se reconocen ocho categorías gramaticales: sustan-

tivo, adjetivo, verbo, adverbio, pronombre, conjunción, interjección y posposición. [18]

Para el caso de estudio que se aborda en el presente informe se centrará la investigación en la gramática de los verbos, que son a nuestro entender la categoría gramatical de mayor complejidad en el guaraní.

2.1.1. Introducción a los verbos en guaraní

Los verbos se clasifican, en dos grandes grupos: verbos propios y categorías léxicas verbalizadas. Esta clasificación se hace en función de la raíz verbal. Los verbos propios tienen raíces verbales y las categorías léxicas verbalizadas generalmente utilizan raíces nominales, pero también pueden tener como raíz adjetivos o adverbios. Por ejemplo, "aguata" (camino) tiene como raíz verbal el verbo "guata" (caminar) por lo que se clasifica como un verbo propio, mientras que "amitã" (soy un niño) tiene como raíz el sustantivo "mitã" (niño) que se verbaliza.

A su vez los verbos propios se pueden clasificar según sus partículas en areales, aireales y chendales.

Los verbos areales son activos, en su gran mayoría de acción mediata o de mayor duración relativa de la acción, por ejemplo, "a-guata" ("aguata"), que en español se traduce como "camin-o" ("camino").

Los verbos aireales son también verbos activos, pero la mayoría, son de acción inmediata, por ejemplo: "rei-kytĩ" ("reikytĩ"), que en español significa "cort-ás" ("cortás").

Los verbos chendales en general se corresponden con la forma atributiva de los verbos ser, estar o parecer. Estos tres verbos generalmente son denominados copulativos en español, tienen una forma atributiva en la cual no tienen un significado relevante, simplemente unen el sujeto al atributo calificativo que es la palabra más importante de una oración con predicado nominal. Es posible reconocerlos al encontrar en la oración una estructura con sujeto, cópula y predicado. El sujeto es el concepto sobre el cual se afirma o niega algo. La cópula es el concepto que relaciona al sujeto con el predicado. Y

el predicado es lo que se afirma sobre el sujeto. Por ejempo, “chekuerai” en español significa “estoy aburrido”, donde el sujeto es la primera persona del singular, la cópula es el verbo “estar” y el predicado es “aburrido”. Llevando el ejemplo al guaraní, el predicado es “kuerai” (“aburrido”), mientras que el prefijo “che” indica primera persona del singular para un verbo chendal. [22] [17]

Cada verbo se forma por un lema y uno o varios morfemas que indican los diferentes accidentes verbales. El lema es el núcleo o raíz del verbo y es lo que le da significado a la palabra.

En los verbos se pueden encontrar 5 accidentes verbales distintos: número y persona (funcionan como un único accidente), forma, tiempo, voz y modo. [32] [18] A partir de los ejemplos presentes en el corpus y de la información recopilada, se logró llegar a la conclusión de que los accidentes verbales tienen un orden de aparición determinado según su tipo. En primer lugar se aplica el accidente de forma. Como se verá más adelante se forma de un prefijo, y en el caso de la negación además incluye un sufijo. El prefijo se ubica al comienzo del verbo y para los casos en los que hay sufijo, éste se ubica luego del sufijo de modo. Luego del prefijo del accidente de forma se agrega el prefijo correspondiente al accidente de número y persona, seguido del de voz. Después se incluye el lema del verbo, y a continuación se agregan los sufijos de modo, el sufijo de forma en caso de que haya, y por último el accidente de tiempo verbal. En la imagen 2.1 se muestra el orden de manera gráfica.

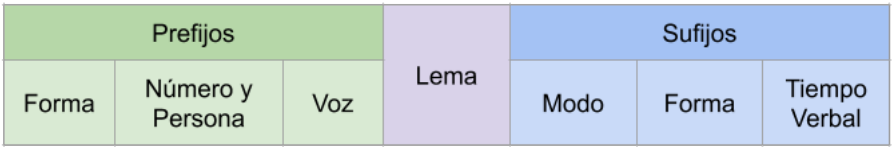

Figura 2.1: Orden de Accidentes verbales

2.1.2. Accidente de Número y Persona

Los números gramaticales son Singular y Plural, están implícitos en las partículas de personas gramaticales. Las partículas de persona gramaticales

se dividen en tres subgrupos Categóricas, Conjugación Optativa y Conjugación Imperativa.

En las partículas categóricas el hablante pone en funcionamiento un verbo indicando simplemente la acción a través del mismo, por ejemplo, “Che aguejy”, en español significa “yo bajo”, dónde “guejy” es el verbo en infinitivo “bajar” que unido al prefijo “a” resulta en el verbo conjugado “bajo”. Las partículas más comunes para este tipo de verbos son: a, re, o (Singulares), ja/ña, ro, pe, o (Plurales).

La conjugación optativa no tiene equivalente en español, el hablante comunica una decisión o una orden a sí mismo. Sus indicadores verbales son: ta, tere, to, taja/taña, toro, tape, to.

La conjugación imperativa se utiliza para dar órdenes personales a la segunda persona gramatical y se conjuga solo en esta persona. Sus partículas son: e y pe. Por ejemplo, “Ekaru”, en español significa “comé!”.

En las tablas 2.2 y 2.3 se muestran las diferentes partículas utilizadas para representar el accidente de número y persona. [32] [18]

Singular

	Prefijo	Acento	Tipo
1ra	a	oral/nasal	areal
	ai	oral/nasal	aireal
	che	oral/nasal	chendal
	ta	oral/nasal	-
2da	re	oral/nasal	areal
	rei	oral/nasal	aireal
	nde	oral	chendal
	ne	nasal	chendal
	tere	oral/nasal	-
	e	oral/nasal	-
	pe	oral/nasal	-
3ra	o	oral/nasal	areal
	oi	oral/nasal	aireal
	i	oral/nasal	chendal
	ij	oral/nasal	-
	iñ	oral/nasal	-
	ho	oral/nasal	areal
	to	oral/nasal	areal

Cuadro 2.2: Accidente de Número y Persona: Singular

Plural

	Prefijo	Acento	Tipo
1ra Incluyente	ja	oral	areal
	ña	nasal	areal
	jai	oral/nasal	aireal
	ñai	oral/nasal	aireal
	ñande	oral	chendal
	taja	oral	-
	taña	nasal	-
1ra Excluyente	ñanea	nasal	areal
	ro	oral/nasal	areal
	roi	oral/nasal	aireal
	ore	oral/nasal	chendal
	toro	oral/nasal	-
2da	pe	oral/nasal	areal
	pei	oral/nasal	aireal
	pende	oral	chendal
	pene	nasal	-
	tape	oral/nasal	-
	pe	oral/nasal	-
3ra	o	oral/nasal	areal
	oi	oral/nasal	aireal
	i	oral/nasal	chendal
	ij	oral/nasal	-
	iñ	oral/nasal	-
	ho	oral/nasal	areal
	to	oral/nasal	areal

Cuadro 2.3: Accidente de Número y Persona: Plural

2.1.3. Accidente de Forma

El accidente de forma indica si la acción del verbo es afirmativa, negativa o interrogativa en la oración. La forma afirmativa no lleva partículas, las otras dos se detallan a continuación. [32] [18]

Negación

La negación indica la no acción del verbo, tiene la particularidad de que se forma mediante una partícula prefija en concordancia con una sufija. Por ejemplo,"nd-aguata-i" (no camino), en este caso se utiliza el prefijo "nd" y el sufijo "i" para negar el verbo "guata" (caminar).

Además, en este tipo de forma se debe tener en cuenta el número y persona a utilizar en la conjugación del verbo dado que las partículas presentan variaciones al referirse a determinados tipos de número y persona.

Otra particularidad que presenta la negación es que las palabras terminadas en "i" agregan a la izquierda del sufijo la letra "r". Por ejemplo, "n-añani-ri" (no corro), se forma por el prefijo "n" y el sufijo "i", pero dado que la palabra termina en "i", el sufijo se transforma en "ri". Cuando se quiere dar énfasis a la negación, se usa el pleonasmo "iri" con palabras no terminadas en "i".

En la tabla 2.4 se muestran las partículas utilizadas para la negación. [32] [18]

Prefijo	Sufijo	Número y Persona	Acento
nd	i	1ra, singular o 3ra	oral
n	i	1ra, singular o 3ra	nasal
nde	i	2da, singular	oral
ne	i	2da, singular	nasal
nda	i	1ra, plural incluyente o 2da, plural	oral
na	i	1ra, plural incluyente o 2da, plural	nasal
ndo	i	1ra, plural excluyente	oral
no	i	1ra, plural excluyente	nasal

Cuadro 2.4: Accidentes de Forma: Negación

Interrogación

Esta forma se utiliza para realizar una interrogación de la acción del verbo. En guaraní no se utiliza la marca de interrogación, las partículas en sí mismas demuestran que se trata de una interrogante. Por ejemplo, para preguntar "¿caminás?", se debe conjugar el verbo "guata" ("caminar") con el prefijo "re" para indicar que el sujeto del verbo es 2da persona del plural ("reguata"), y luego agregar el sufijo "pa" para indicar que es una pregunta, quedando como resultado la palabra "reguatápa". Además la expresión no se entona interrogativamente al ser pronunciada. En la tabla 2.5 se muestran los sufijos utilizados para los verbos interrogativos. [18][32]

Sufijo	Acento
pa	oral/nasal
piko	oral/nasal
tiko	oral/nasal
tepa	oral/nasal

Cuadro 2.5: Accidentes de Forma: Interrogación

2.1.4. Accidente de Voz

Establece la relación entre el sujeto y la acción del verbo en una oración. Pueden ser de dos clases: voz pasiva y voz activa.

La voz pasiva expresa que el sujeto recibe los efectos de una acción. Se utiliza el prefijo “je” para los verbos orales y “ñe” para verbos nasales. Por ejemplo, “añekytĩ” en español significa “me corté”, se forma por el prefijo “a” que indica la persona que recibe la acción del verbo (“yo”), seguido del prefijo “ñe” que correspondiente con la voz pasiva indicando que el sujeto recibe los efectos de la acción, y no es quien la ejecuta. Finalmente se agrega el lema “kytĩ” que significa “cortar”.

La voz activa refleja que el sujeto realiza la acción. Un ejemplo de verbo en voz activa es “aguata” (“camino”) que fue analizado anteriormente, dónde el sujeto es el que realiza la acción de caminar. La voz activa se subdivide en varias sub-clases que se detallan a continuación, pero el uso mas frecuente es la denominada voz activa simple, en la cual no se agregan partículas.

En la tabla 2.6 se especifica la clasificación para el accidente de voz.

Voz	Prefijo	Acento	Descripción
Pasiva	je	oral	Muestra que se recibe los efectos de una acción.
	ñe	nasal	
Activa Simple	-	oral/nasal	Refleja que el sujeto realiza la acción.
Activa Coactiva	mbo	oral	El sujeto oficia de agente indirecto y manda a realizar la acción a otra persona.
	mo	nasal	
Activa Reciproca	jo	oral	Necesita de dos sujetos, y se conjuga solamente en plural.
	ño	nasal	
Activa Objetiva	poro	oral/nasal	El sujeto realiza la acción en personas indeterminadas.
	mba'e	oral/nasal	
Activa Subsuntiva	guero	oral/nasal	El sujeto realiza la acción para conseguir que el paciente realice la misma acción.
	ro	oral/nasal	

Cuadro 2.6: Accidentes de Voz

2.1.5. Accidente de Tiempo

Indica el tiempo en que se realiza la acción del verbo. Los tiempos son: presente, pasado y futuro, pero a su vez existen sub-clases para cada uno de ellos, las mismas se detallan en la tabla 2.7. Al igual que en el español el tiempo presente representa una acción que se realiza en el momento actual, pasado referencia a que la acción se realizó previamente, y futuro que la acción se desarrollará posteriormente al momento actual. [32][18]

Tiempo	Sufijo	Descripción
Presente	-	La acción se realiza en el momento actual.
Presente Perfecto	ína	Indica una acción actual que continúa.
Presente intermitente	ikóni	Indica una acción actual, cuya realización se produce con interrupciones o intermitencias.
Pretérito perfecto	akue	Indica que la acción se realizó en un tiempo pasado.
Pretérito pluscuamperfecto	va'ekue	Indica que la acción se realizó en un tiempo más lejano al pretérito.
Pretérito imperfecto	mi	Expresa una acción pasada cuyo principio y fin no está definido.
Pretérito reciente	kuri	Señala que la acción se realizó en un tiempo pasado reciente.
Pretérito remoto	raka'e	Indica una acción en tiempo lejano.
Pretérito anterior	ra'e	Indica un tiempo relativamente reciente.
Futuro perfecto	ta	Indica una acción a realizarse, sin precisar el momento.
Futuro obligatorio	va'erã	Indica una acción que deberá realizarse de modo imprescindible.
Futuro necesario	arã	Indica una acción que debe realizarse en el futuro, por necesidad.
Futuro próximo	pota (oral) mbota (nasal)	Indica que la acción va a realizarse en un momento cercano, en el futuro.
Futuro dudoso	ne	Indica una acción que se realizará dudosamente, en algún momento futuro.

Cuadro 2.7: Accidentes de Tiempo

2.1.6. Accidentes de Modo

El accidente de modo indica la manera en que se realiza la acción del verbo. Se dividen en dos grandes tipos: Indicativos e Imperativos, que a su vez se sub-dividen en otras categorías como se detalla en las tablas 2.8, 2.9, 2.10 y 2.11. [32][18]

Indicativos

El modo indicativo expresa una acción concreta y objetiva. Puede tratarse de una acción que transcurre en el momento, que ya ha ocurrido o que está por acontecer. Por ejemplo, "opurahéijoa" significa "cantan en coro", "o" referencia al sujeto que realiza la acción en este caso es 3ra persona del plural, luego se concatena el lema "purahéi" que significa cantar, y por último se agrega el sufijo "joa" que se utiliza para representar el modo indicativo colectivo que indica que la acción se realiza de manera colectiva.

Modo	Sufijo	Descripción
Indicativo Simple	-	Expresa la actitud objetiva del hablante con respecto al verbo.
Indicativo Volitivo	se	Expresa el deseo de que se realice la acción.
Indicativo Supositivo	po, nipo, pipo, poku	Implica una suposición de que la acción se realice.
Indicativo Conjetural	ramo, rõ, rire	Expresa la falta de seguridad de que la acción se realice.
Indicativo Condicional	mo	Indica lo que hubiera sido hecho.
Indicativo Habitual	va	Indica que la acción se realiza en forma acostumbrada.
Indicativo Totalitivo	pa, mba	Indica que la acción se desarrolla en su totalidad
Indicativo Cuasiaccional	vy, ngy, ky	Indica que la acción se desarrolla a medias, no en su totalidad

Cuadro 2.8: Accidentes de Modo: Indicativo

Modo	Sufijo	Descripción
Indicativo Mediativo	uka	Indica que la acción se realiza a través de otra persona.
Indicativo Locativo	ha	Hace referencia al lugar de la acción del verbo.
Indicativo Factivo	hára, ha	Indica la actividad o función habitual del agente, si es persona, señala su profesión con la partícula.
Indicativo Concomitante	vo	Indica la simultaneidad de la acción.
Indicativo Proximal	nunga	Indica que la acción es medianamente realizada.
Indicativo Colectivo	joa	Indica que la acción es realizada por dos o más personas.
Indicativo Narrativo Verosímil	niko, ningo, ko, ngo	Se resalta la veracidad de lo que se dice.
Indicativo Narrativo Inverosímil	ndaje, je	Se resalta la no veracidad de lo que se dice.
Indicativo Anhelativo	nga'u	Expresa el anhelo de que se realice la acción.
Indicativo Simulativo	gua'u	Indica que la acción se realiza aparentemente, pero en realidad, no es así. Acción completamente simulada o fingida.
Indicativo Aparencial	vaicha	Indica una acción aparente.
Indicativo Pietativo	anga	Indica una acción por cuyo autor se siente piedad o conmiseración.
Indicativo Frecuentativo	mante	Indica que la acción se realiza en forma frecuente.
Indicativo de Opción Única	mante	Indica una acción realizada bajo cierto condicionamiento que no ofrece alternativa.
Indicativo Frecuentativo Enfático	manterei	Indica que la acción se realiza en forma muy frecuente.
Indicativo Frustrativo	rei	Refiere la inutilidad de la acción.
Indicativo Intencional	mo'ã	Indica que se tenía la intención de que la acción se realice, aunque no se realizó.

Cuadro 2.9: Accidentes de Modo: Indicativo

Modo	Sufijo	Descripción
Indicativo Antelativo	jepe	Indica que la acción se realizó en forma previa.
Indicativo Aseverativo	ma	Indica la realización de la totalidad de la acción.
Indicativo Inmotivado	nte	Indica que la acción se realiza sin motivo consciente.
Indicativo Usual/Ocasional	jepi	Indica que la acción del verbo se realiza en forma acostumbrada.
Indicativo Confirmativo	katu	Indica que una acción verbal anunciada se ha cumplido.
Indicativo Intermitente	mimi	Indica que la acción se desarrolla con breves intervalos de tiempo.
Indicativo Dubitativo	nune	Indica una acción de cuya realización tiene dudas el hablante, pero presume que se ha producido.
Indicativo Persistencial	meme	Indica una acción realizada y repetida continuamente
Indicativo de acción leve	piguy	Indica un hecho levemente notado.
Indicativo Intermediativo	rupi	Indica una acción que ha servido como medio para lograr o evitar algo.

Cuadro 2.10: Accidentes de Modo: Indicativo

Imperativos

El imperativo es un modo verbal que se usa para expresar órdenes, pero también mandatos, ruegos e incluso deseos. Por ejemplo,"aguatáva" que significa "suelo caminar" es un verbo con modo imperativo habitual. Se compone del prefijo "a" que representa el sujeto que realiza la acción en este caso es 1ra persona del singular (yo), seguido del lema "guata" (caminar), y por último se le agrega el sufijo "va" que indica el modo Imperativo Habitual del verbo. El Imperativo Simple indica una orden para que se realice la acción. No tiene partícula sufija y se conjuga con las partículas de número y persona: "e" o "pe" para la segunda persona del Singular y Plural y con las partículas de la conjugación optativas: "ta", "tere", "to", "taja", "taña", "toro", "tape" y "to", para las demás personas.

Modo	Sufijo	Descripción
Imperativo Simple	-	Empleado para expresar mandatos, órdenes, solicitudes, ruegos o deseos.
Imperativo Conminativo/ Categórico	ke	Indica orden o mandato.
Imperativo Rogativo	na	Indica un ruego de que se realice la acción del verbo.
Imperativo Amistoso	mi	Indica un pedido amable de que se realice la acción del verbo/nasal.
Imperativo Compuesto	míkena	Enfatiza el pedido.
Imperativo Indeterminado	mba'e	-
Imperativo Permisivo	katu	Expresa que la acción se ha realizado afirmativamente/nasal.
Imperativo Incisativo	py	Es un imperativo formado por vía de hispanismo.
Imperativo Preventivo	mandi	Indica una acción que debe realizarse sin pérdida de tiempo y antes que cualquier otra cosa ocurra, adelantándose a probables inconvenientes.

Cuadro 2.11: Accidentes de Modo: Imperativo

2.2. Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural (PLN) es una rama de la inteligencia artificial que mediante un conjunto de métodos y técnicas eficientes logran que las computadoras "aprendan" a entender o generar el lenguaje natural. El objetivo principal de PLN es leer, descifrar, comprender y dar sentido a los idiomas humanos. Es considerado un problema muy difícil debido a la complejidad inherente del lenguaje humano. Por ejemplo, cuando se utiliza el sarcasmo, es muy difícil que un sistema "aprenda" mediante reglas que es lo que realmente se quiso decir. Otro de los problemas que presentan los lenguajes naturales es la ambigüedad y la imprecisión que tienen algunos términos. La comprensión integral del lenguaje requiere comprender las palabras y la forma en que los conceptos están conectados para transmitir el mensaje deseado. [3]

Las aplicaciones relacionadas son innumerables, se presentarán a continuación algunas de las más relevantes.

- **Análisis de Sentimientos:** es el proceso de determinar el tono emocional que hay en las oraciones. Es utilizado para entender las actitudes, opiniones y emociones en la oración. Actualmente es muy utilizado para la monitorización de las redes sociales, permite entender una idea general de la opinión pública sobre temas específicos. [4]
- **Recuperación y extracción de información:** consiste en recopilar información de bases de documentos, o cualquier tipo de documentos electrónicos para generar y almacenar información estructurada que luego se puede utilizar con el objetivo de consultar o recuperar textos, imágenes, sonidos o datos de otras características. La extracción de información resulta esencial para clasificar, resumir y encontrar información a través de Internet. Esta técnica de PLN se utiliza mucho en los buscadores online que requieren recuperar páginas web relacionadas a palabras claves. [25] [6]
- **Respuesta a preguntas:** tiene como objetivo responder mediante computadora las preguntas realizadas en lenguaje natural, encontrando la relación entre preguntas y respuestas. Se intenta reconocer preguntas

del tipo “cómo”, “dónde”, “por qué”, definiciones, listas, etc. Los sistemas como Siri, Google Assistant o asistentes virtuales son ejemplos de este tipo de técnicas. Deben ser capaces de responder de forma verbal o escrita las preguntas realizadas por el usuario, incluso en algunos casos permitir una conversación fluida.

- **Traducción Automática de Textos:** dada la existencia de miles de idiomas surge la necesidad de tener traducciones de los textos que permitan acceso globalizado a la información. La traducción automática es una técnica de PLN enfocada a esta tarea, se detallará más sobre esta aplicación en la sección 2.3.
- **Resúmenes de textos automáticos:** una de las aplicaciones puede ser decidir si el contenido de textos de diferentes páginas web es relevante. Para esto se analiza el documento y se genera un resumen, esto permite que los lectores interesados en un tema tengan una idea rápida del contenido del mismo [6]

Las técnicas basadas en PLN facilitan la comunicación con las computadoras, permitiendo automatizar procesos que se realizan manualmente como la traducción o clasificación de documentos, lo que genera un ahorro significativo de tiempo. Por otro lado, ayuda en la toma de decisiones, por ejemplo, tener una idea general de opiniones en redes sociales puede ayudar a detectar y prevenir acontecimientos sociales, permitiendo actuar de manera rápida y efectiva. [6]

El procesamiento del lenguaje natural generalmente sigue una secuencia de pasos en su aplicación para la resolución de problemas. Estos pasos son pre-procesamiento, análisis morfológico, análisis sintáctico, análisis semántico y análisis pragmático. A continuación se abordaran brevemente las etapas más relevantes de este proceso. [26]

2.2.1. Pre-procesamiento

El pre-procesamiento puede incluir las siguientes actividades: extracción de textos relevantes, detección de idioma y tokenización.

La extracción de textos relevantes consiste en filtrar la información relevante de los textos, descartando el ruido que puedan tener. Por ejemplo, el contenido obtenido de páginas de web, puede introducir mucha información irrelevante que sólo genera ruido en el corpus como imágenes, tablas u otros elementos que no aportan a la comprensión del lenguaje. Se debe seleccionar el contenido relevante de cada texto y guardarlo en un repositorio donde queda todo el material relevante almacenado.

La detección del idioma consiste en reconocer el idioma del texto en caso de que no sea conocido. En algunos casos ocurre que un mismo texto tiene fragmentos en diferentes idiomas que deben ser correctamente identificados y clasificados.

La tokenización consiste en organizar el texto identificando los tokens, es decir, las palabras y otros elementos del texto que puedan resultar relevantes (por ejemplo números, fechas y signos de puntuación). Cada palabra queda guardada en una estructura denominada token.

En algunos casos se genera un índice o base de palabras que contiene todas las palabras que tiene el texto y la información necesaria para localizarla dentro del mismo. La dificultad en este paso se centra en poder distinguir las palabras de los términos multi-palabras, fechas, nombres propios, siglas, abreviaturas e incluso errores. [26]

2.2.2. Análisis Morfosintáctico

Esta etapa consiste en etiquetar gramaticalmente cada palabra incluida en los tokens y realizar el análisis morfológico.

El etiquetado gramatical también conocido como POS-tagging consiste en etiquetar cada palabra con su correspondiente categoría gramatical (nombre, verbo, adverbio, etc). Este proceso se detallará en la sección 2.2.2.1.

Por otro lado el análisis morfológico consiste en analizar la estructura interna de la palabra para descubrir sus atributos morfológicos como género, número, tiempo verbal, modo, etc. Permite obtener una forma canónica para cada palabra. Este etiquetado se realiza con un analizador morfológico que

es una herramienta utilizada para obtener los morfemas de las palabras. Los morfemas son las unidades más pequeñas del idioma que tienen significado léxico o gramatical y no se pueden dividir en unidades significativas menores.

Normalmente esta tarea se apoya en el etiquetado gramatical realizado previamente, aunque a veces se averigua la categoría gramatical y se realiza el análisis morfológico al mismo tiempo. El principal problema en esta etapa es que muchas veces un mismo lema puede ser etiquetado de varias formas, para desambiguar se requiere conocimiento del contexto en el que es utilizado el lema. Generalmente se utilizan desambiguadores morfológicos que a veces solucionan por completo la ambigüedad, y en otras ocasiones descartan las opciones menos probables. Los desambiguadores pueden ser basados en reglas, estadísticos o híbridos.

A modo de ejemplo se realizará el análisis morfológico de la palabra "omoheñóikuri" (en español significa "originó"), en este caso la palabra se descompone en "o-moheñói-kuri", donde "o" indica 3ra persona, "moheñói" es el lema del verbo que en español significa "originar" y se clasifica como verbo, y "kuri" indica pretérito reciente.[26]

2.2.2.1. POS tagging

Part-of-speech tagging (POS tagging), también conocido como etiquetado gramatical, es el proceso de asignar una categoría gramatical a cada una de las palabras de un texto. Este proceso puede ser realizado según la definición de la palabra o según el contexto en que aparece.

Las categorías gramaticales aportan gran cantidad de información acerca de una palabra y de sus palabras adyacentes en el texto. Conocer la categoría gramatical de una palabra es de gran utilidad ya que permite saber de qué forma debe ser interpretada.

Sin embargo, uno de los inconvenientes que presenta el POS tagging es la ambigüedad, algunas palabras pueden pertenecer a más de una categoría gramatical. Se debe recurrir al contexto en que se encuentra la palabra para poder discernir entre las posibles categorías gramaticales cuál es la correcta, esto a veces puede resultar complejo. Este problema habitualmente ocurre en

el lenguaje natural dado que posee una gran cantidad de palabras ambiguas, al contrario de lo que sucede con los lenguajes artificiales. Un ejemplo para el idioma español es la palabra “coma” que dependiendo del contexto puede referir al acto de ingerir alimentos siendo su categoría gramatical el verbo o al signo ortográfico siendo su categoría gramatical el sustantivo.

A modo de ejemplo se realiza el POS tagging de una frase en guaraní y de su traducción al español. Las frases fueron extraídas del curso guaraní-español de la herramienta Duolingo[1].

Peteî	mitâ	ho’u	yva
adjetivo	sustantivo	verbo	sustantivo

Un	niño	come	fruta
artículo	sustantivo	verbo	sustantivo

Al alinear las oraciones, se puede observar que “Peteî” y “Un” se corresponden. Sin embargo, en la gramática guaraní el artículo no existe como categoría léxica. Por lo tanto en la frase en guaraní se está utilizando el adjetivo “peteî” mientras que para la frase en español se utiliza el artículo indefinido “un”.

Existen dos grandes grupos para el etiquetado léxico: “Aproximaciones Lingüísticas” y “Aproximaciones de Aprendizaje Automático”. También existen “Aproximaciones Híbridas” que combinan algunos aspectos de las aproximaciones anteriores. [5]

Las “Aproximaciones Lingüísticas” se basan en el conocimiento lingüístico, normalmente descripto mediante un conjunto de reglas establecidas de forma manual o aprendidas de forma semiautomática. Este tipo de aproximaciones fueron una de las primeras soluciones para resolver la ambigüedad léxica. Los primeros etiquetadores consistían en un conjunto de reglas escritas manualmente por lingüistas con el fin de predecir las posibles categorías gramaticales de una palabra. TAGGIT fue el primer etiquetador que podía

[1] www.duolingo.com

ser aplicado a una gran cantidad de texto [20]. Este luego fue utilizado para la construcción de grandes corpus como Brown.

Este tipo de sistemas presentan varios problemas, entre ellos que requieren un gran costo humano para definir las reglas y que solamente sirven para la lengua para la que se han construido. Otra gran desventaja es que si se dispone de un léxico muy reducido, muchas situaciones no son tenidas en cuenta causando que los casos de ambigüedad contemplados sean pocos y como consecuencia dificulte la exportación de las reglas a otras lenguas.

A pesar de los inconvenientes que presenta este tipo de aproximación, al construir modelos de lenguaje desde un enfoque lingüístico es posible incorporar muchas y complejas fuentes de información provocando que sean más expresivas, presentando de esta forma mejores desambiguaciones que otros tipos de aproximaciones.

Las "Aproximaciones de Aprendizaje Automático" elaboran un modelo de lenguaje usando métodos de aprendizaje a partir de datos, generalmente utilizan corpus anotados con información lingüística. Existen varios métodos de aprendizaje entre ellos Markov Model o n-gramas, redes neuronales, árboles de decisión, reglas de transformación y autómatas. Una de las aproximaciones más usada es la de modelo oculto de Markov o también conocido como HMM por sus siglas en inglés, Hidden Markov Model. [10] [12]

Un modelo estadístico con HMM aplicado a POS tagging pretende modelar una oración como un proceso de Markov en el cual se obtiene como resultado las categorías gramaticales de cada una de las palabras de la oración. En este proceso los estados observables son las palabras de la oración y las etiquetas gramaticales son estados ocultos que se decodifican como resultado del método. Esta es una de las técnicas que se utilizarán en el desarrollo de este trabajo. Se dispone de más información sobre esta técnica en el anexo A.

2.2.3. Análisis Sintáctico

El análisis sintáctico tiene como objetivo analizar cómo las palabras se combinan para formar construcciones gramaticalmente correctas.

Una opción para realizar este análisis es generando un árbol sintáctico que consiste en una estructura en árbol con las categorías sintácticas formadas por cada una de unidades léxicas que aparecen en la oración. Las categorías o componentes sintácticos que aparecen en la oración se denominan sintagmas, son los grupos de palabras que constituyen una unidad sintáctica, cumplen una función determinada con respecto a otras palabras de la oración.

Por ejemplo, se puede realizar un análisis sintáctico de la siguiente oración "che aguata ekuélape", donde el pronombre "che" ("yo") es el sujeto de la oración y corresponde a un sintágma nominal. Seguido del sujeto se encuentra el verbo "aguata" ("camino") y el complemento "ekuélape"("a la escuela"). Este último se podría clasificar como un sintagma preposicional en español, pero en guaraní se calificaría como sintagma posposicional. En la imagen 2.2 se muestra el árbol sintáctico que se genera del análisis anterior. [26]

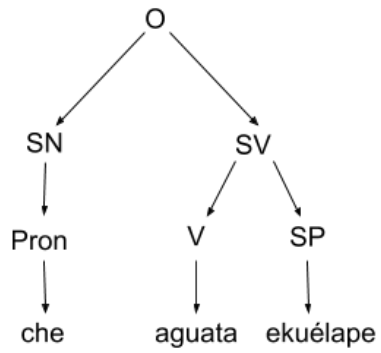

Figura 2.2: Árbol Sintáctico para la oración "che aguata ekuélape"

Otra forma de realizar el análisis es mediante los árboles de dependencia. Estos establecen relaciones de dependencia entre las palabras, pueden ser orientados a la sintaxis (sujeto, objeto directo, objeto indirecto, determinante, etc) o a la semántica identificando diferentes roles (agente, tema, etc).

En la figura 2.3 se muestra el análisis de dependencias correspondiente a la oración que se ha estado analizando. En la misma se puede ver que "che" oficia de sujeto del verbo "aguata", mientras que "ekuélape" es un complemento del verbo. [26]

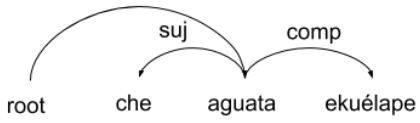

Figura 2.3: Análisis sintáctico para la oración “che aguata ekuélape”

2.2.4. Análisis Semántico

El análisis semántico se enfoca en el estudio del significado de las oraciones, el objetivo es generar una estructura semántica. La semántica hace referencia a la condición de verdad de la oración, independientemente del contexto, cada palabra tiene un significado que no depende de la intención del hablante y la influencia del contexto. El significado de una oración en un contexto específico es lo que se estudia en el análisis pragmático, que sería la siguiente etapa del análisis.

2.3. Traducción automática

La traducción automática se define como la traducción de un lenguaje de origen a uno de destino que se realiza sin intervención humana. Es uno de los primeros problemas que se intentaron resolver con PLN, sus primeros pasos se remontan a la década del 50, sin embargo los avances más significativos surgen alrededor del año 2000, con la globalización del Internet. Es un campo cada vez más amplio, y que se sigue expandiendo, sin embargo aún sigue siendo un problema muy difícil para lenguas con escasos recursos lingüísticos como es el caso del guaraní. No alcanza con una sustitución palabra a palabra, un traductor debe interpretar y analizar todos los elementos del texto y saber cómo cada palabra puede influir en otra, se requiere conocimiento en gramática, sintaxis (estructura de oraciones) y semántica (significados) en los idiomas de origen y destino.

Para realizar esta traducción se utilizan métodos basados en reglas (RBMT, Rule Based Machine Translation), métodos estadísticos (SMT, Statistical Machine Translation) o redes neuronales (NMT, Neural Machine Translation).[1] [33]

2.3.1. Métodos Basados en Reglas

Los métodos basados en reglas parten de herramientas y/o recursos lingüísticos para crear una traducción. Se basan en reglas lingüísticas y gramaticales del idioma, así como en diccionarios con palabras comunes del idioma, por lo cual se requieren extensos léxicos con información morfológica, sintáctica, semántica, y grandes conjuntos de reglas. Estas reglas se utilizan para luego transferir la estructura gramatical del idioma de origen al idioma de destino.

Los resultados de este tipo de traducción pueden carecer de la fluidez que los lectores esperan, sin embargo suelen generar traducciones coherentes, y predecibles.

Por otro lado, este tipo de traducción automática requiere de una gran inversión en tiempo y recursos. Para desarrollar un traductor se requiere

contar con gramáticas del idioma origen y destino, así como diccionarios y reglas de transferencias.

Otra desventaja de este método es que no puede traducir estructuras lingüísticas que no estén en la gramática del modelo o en sus reglas de transferencias, tampoco puede reconocer palabras que no se encuentren en sus diccionarios. Por lo tanto, el mantenimiento de estos métodos debe ser casi continuo para asegurar que traduzcan textos nuevos o estructuras que no estén previstas.

En muchas ocasiones los métodos basados en reglas funcionan como facilitador para la construcción de traductores de idiomas similares, como es el caso del español y el catalán. Sin embargo, los idiomas generalmente son muy distintos, por lo que el modelo utilizado para la traducción en un determinado idioma no es aplicable a otros. Esto se debe a que es un método estrictamente dependiente de la gramática, sintaxis y semántica de cada idioma.

Una de las principales ventajas de este método frente a la traducción estadística y basada en redes neuronales es que no requiere de un gran corpus para ser entrenado. En el caso del idioma guaraní el corpus existente es muy pequeño por lo que parecería una mejor opción frente a los métodos estadísticos. Sin embargo los métodos basados en reglas requieren muchos recursos lingüísticos como parsers, o herramientas que faciliten la programación de las reglas gramaticales y estos recursos también son muy escasos para el idioma guaraní. [1] [33] [13]

2.3.2. Métodos Estadísticos

Los métodos estadísticos se basan en modelos de traducción construidos en base a textos de origen y textos de destino con la traducción correspondiente. Este tipo de métodos no tiene ningún conocimiento de normas lingüísticas, las traducciones se realizan a partir de grandes volúmenes de datos en ambos idiomas. Esta es una de sus principales ventajas frente a los modelos basados en reglas, solo se requieren datos para poder entrenarlos. Concretamente se requiere un corpus paralelo con texto en el idioma de origen y su correspondiente traducción en el idioma de destino.

Los modelos estadísticos tradicionales se forman de tres grandes elementos: el modelo de lenguaje, el modelo de traducción, y un decodificador. El modelo de lenguaje es el componente encargado de calcular las probabilidades de que una oración sea correcta, para esto se utiliza el corpus en el idioma destino. El modelo de traducción establece la correspondencia entre el idioma de origen y el idioma de destino, este modelo se entrena con el corpus paralelo. Durante el entrenamiento se estima la probabilidad de una traducción a partir de las traducciones que aparecen en el corpus de entrenamiento. El decodificador es el componente encargado de encontrar las traducciones candidatas más probables entre todas las traducciones posibles. Por lo general devuelve entre 100 y 500 traducciones posibles. Por lo cual, dado un modelo de lenguaje y un modelo de traducción, se crean algunas de las traducciones posibles y se elije la más probable.

Los modelos de traducción estadística utilizan parámetros que provienen del análisis del corpus. La construcción de modelos de traducción estadística es un proceso rápido, pero la tecnología depende en gran medida de los corpus paralelos existentes. Se requiere de millones de palabras para un dominio específico y aún más para el lenguaje general.

Además, la traducción automática estadística requiere una gran cantidad de procesamiento y una amplia configuración de hardware para ejecutar modelos de traducción para un rendimiento promedio. Por lo general, los sistemas estadísticos ofrecen traducciones menos coherentes. Proporcionan buena calidad cuando hay corpus grandes y calificados disponibles. Sin embargo, la traducción no es predecible ni consistente. Además, se requiere un hardware significativo para construir y gestionar grandes modelos de traducción. [1] [33] [8]

2.3.3. Redes Neuronales

Los métodos neuronales son un subconjunto de métodos estadísticos que han ganado mucha notoriedad en la actualidad. Sin embargo se enmarcan en una sección diferente dado que la lógica que implementan es muy distinta a la que utilizan los métodos estadísticos clásicos. Las redes neuronales también requieren grandes corpus paralelos, muchas veces requieren un ma-

yor volumen de datos que los modelos estadísticos clásicos. Estos sistemas están inspirados en el comportamiento de las neuronas biológicas del cerebro humano, con el correr del tiempo se han transformado en una herramienta matemática para el modelado de funciones. Al igual que las neuronas reciben información y realizan conexiones entre sí, los componentes del lenguaje se asocian con otra información subyacente para formar asociaciones y generar traducciones. Utilizando técnicas de aprendizaje automático, el sistema aprende a traducir a partir de grandes cantidades de textos paralelos. Cada palabra junto con toda su información asociada se utiliza para entrenar el modelo. Por ejemplo, si tenemos las palabras "perro", "gato" y "mesa". La palabra "perro" se puede asociar con las palabras "animal", "sustantivo" y "ladra". Por otro lado, la palabra "gato" se puede asociar con las palabras "animal", "sustantivo" y "maúlla". Mientras que "mesa" se puede asociar a "sustantivo" y "madera". El sistema puede aprender, que la palabra "perro" y "gato" son más similares entre sí que "perro" y "mesa", ya que en el texto generalmente están rodeados de palabras similares que induce a pensar en traducciones similares.

2.3.3.1. OpenNMT

OpenNMT (Open-Source Neural Machine Translation) es un framework *open source* implementado en Torch para el entrenamiento de modelos de redes neuronales para la traducción automática de idiomas. El sistema es sucesor de *seq2seq-attn* (Sequence-to-Sequence Learning with Attentional Neural Networks) desarrollado en Harvard, y ha sido reescrito por completo para obtener una mayor eficiencia, legibilidad y generalización.

El sistema principal está desarrollado en el framework matemático Lua/-Torch el cual puede ser fácilmente extendido usando los componentes estándar internos de red neuronal de Torch. También fue extendido para ser soportado por Python/PyTorch con la misma API.

Esta librería requiere tener un corpus paralelo en idioma de origen y destino. De este corpus se utiliza una parte para entrenamiento (alrededor del 80 %), otra para test y otra para validación (como máximo se sugiere 5,000 oraciones). El conjunto de validación es utilizado para evaluar cada paso de la iteración. El primer paso que se realiza es el pre-procesamiento del

corpus de entrenamiento y validación, mediante el cual se obtienen archivos con el vocabulario presente en el corpus. Luego se realiza el entrenamiento del corpus, el modelo base por defecto ejecuta 100,000 iteraciones y cada 5,000 se realiza un punto de validación. Después de realizado el entrenamiento del modelo, se puede realizar traducciones, que luego se pueden evaluar con alguna medida como por ejemplo, BLEU. [14] [16] [21]

2.4. Revisión de antecedentes

Previo al inicio del desarrollo del proyecto, se realizó una investigación acerca de herramientas y recursos lingüísticos relacionados al guaraní, especialmente en la variante jopará, con el objetivo de conocer con qué recursos lingüísticos se puede contar. Existen varios diccionarios en línea que permiten realizar traducciones palabra a palabra, e incluso algunos permiten traducir frases de español a guaraní.

A continuación se detallarán los principales recursos utilizados como base para el desarrollo de la investigación.

Duolingo, es un sitio web en el cual se puede encontrar el significado de las palabras, su uso en oraciones, e incluso comentarios de otros usuarios acerca de la traducción.

Descubrir Corrientes [32] es una página web que cuenta con un diccionario español-guaraní y guaraní-español. Además incluye información acerca de la gramática del lenguaje aplicados a ejemplos.

Glosbe[2] en un sitio web que contiene un diccionario y además cuenta con una base de memoria de traducción que utiliza datos de segmentos traducidos.

iGuarani[3] es una página web en la que se permite traducir palabra a palabra, o realizar traducciones de oraciones de algún idioma a elección (por ejemplo, español) a Guaraní.

SENATICs[4] provee un traductor palabra a palabra de español-guaraní y viceversa.

Por otro lado, se utilizó un corpus paralelo español-guaraní que cuenta con alrededor de 14,500 oraciones extraídas de noticias, cuentos y blogs [9].

En cuanto a la información relacionada estrictamente a la gramática del

[2]https://es.glosbe.com/
[3]http://www.iguarani.com/
[4]https://www.senatics.gov.py/traductor/traducir.php

idioma, se consulta la gramática oficial publicada por la Academia de la Lengua Guaraní [18].

Los traductores encontrados funcionan con una técnica de palabra a palabra y con un vocabulario bastante bajo, por lo que se decidió construir una línea base de traducción desde cero con los recursos encontrados e intentar mejorarla. La mayoría de los recursos existentes no son computacionalmente usables de manera directa. Tampoco se encontraron muchos desarrollos de herramientas particulares de PLN para el idioma guaraní.

Capítulo 3

Desarrollo de la investigación

Este capítulo describe el desarrollo de la investigación del proyecto. En principio, se realizó una investigación exhaustiva de la gramática y de los recursos existentes acerca del lenguaje guaraní. Se abordaron cuatros grandes objetivos: la creación de una base de datos a partir de recursos lingüísticos, la detección de verbos en oraciones, el análisis morfológico de verbos y la traducción automática. Al mismo tiempo se desarrolló una herramienta web con una interfaz para usuarios que detecta verbos en una oración y realiza el análisis morfológico de lo verbos aplicando un método basado en reglas. En este capítulo se desarrollarán las diferentes soluciones propuestas para abordar los problemas mencionados.

3.1. Corpus

La investigación se realizó en base a un corpus paralelo (español - guaraní) [9] que contiene alrededor de 14,500 pares de oraciones alineadas y posee 228,000 tokens guaraníes y 336,000 tokens españoles.

El corpus está conformado por artículos (blogs, cuentos y noticias) de diferentes sitios web paraguayos, los cuales contienen la versión tanto en guaraní como en español. Los artículos de noticias fueron escritos entre diciembre de 2017 y agosto de 2019, predominan en cantidad frente a los otros artículos pero presentan más ruido en su traducción.

Para poder realizar la evaluación de la clasificación y el análisis de los verbos con medidas como Recall, Precision y F1Score o poder entrenar con métodos estadísticos fue necesario hacer un análisis manual de una parte significativa del corpus para saber cuáles palabras son verbos y cuál es su análisis morfológico. Este análisis manual es un proceso que requiere de mucho tiempo por lo que se utilizó sólo una parte del corpus sobre la cual se realizó el análisis manual. Para esto se utilizan todos los archivos de blogs y cuentos dado que es un texto menos ruidoso, con mayor cantidad de palabras en guaraní y menos hispanismos. Se formaron tres conjuntos de archivos, el conjunto entrenamiento con 28 archivos, el conjunto de desarrollo de 12 archivos y el conjunto de test que tiene 13 archivos. El conjunto de entrenamiento fue utilizado para entrenar los diferentes modelos. El conjunto de desarrollo fue utilizado para testear los métodos de manera inmediata e ir ajustándolos y corrigiendo errores en la implementación y en el análisis manual. El conjunto de test[1] fue utilizado para realizar la evaluación de los diferentes métodos. Estos archivos fueron analizados uno a uno detectando en cada oración las palabras que se correspondían con verbos y realizando el análisis morfosintáctico de las que fueran verbos. De esta forma se etiquetaron 1,015 verbos en el conjunto de entrenamiento, 412 en el conjunto de desarrollo, y 477 en el conjunto de test.

Se obtuvieron algunas métricas adicionales para observar las dimensiones de los conjuntos analizados. En la tabla 3.1 se muestra la cantidad de

[1]El conjunto de test fue etiquetado por el tutor del proyecto de manera de tener un conjunto de evaluación externo que no hubiera sido visto durante el desarrollo.

oraciones y palabras distintas que tiene cada subconjunto.

Métrica		Conjunto		
		Entrenamiento	Desarrollo	Test
Cantidad de palabras distintas	Español	2,476	1,185	1,238
	Guaraní	2,502	1,264	1,265
Cantidad de oraciones	Español	471	276	299
	Guaraní	471	276	299

Cuadro 3.1: Métricas de los conjuntos

3.2. Diccionario

Se implementó una base de datos relacional en PostgreSQL [24] con el objetivo de tener un diccionario de palabras en guaraní con la información estructurada. El mismo permite tener acceso rápido a las palabras del idioma, y sus posibles definiciones. Además, es un recurso fácilmente expansible. La construcción de esta base de datos se realizó en base al diccionario publicado en la página Descubrir Corrientes [32]. Este diccionario fue scrapeado mediante el uso de la librería Scapry [28] y guardado en una base de datos local. Previo a su utilización fue necesario realizar correcciones de errores, resolver recursiones en definiciones, reconocer nomeclatura utilizada en el diccionario entre otros ajustes necesarios para la correcta utilización del mismo. Por ejemplo, la palabra "máa" tiene como definición "Ver: máva.", "máva" significa "alguien" o "quién" por lo que es necesario utilizar esta definición para darle un significado a la palabra original. Cada palabra puede tener una o más definiciones asociadas, a su vez cada una de estas definiciones está asociada a una o más abreviaturas. Por ejemplo, la abreviatura "v" indica un verbo, "v. air." indica un verbo aireal, "adj." representa un adjetivo, etc.

En la figura 3.1 se muestra un esquema de la información estructurada en la base.

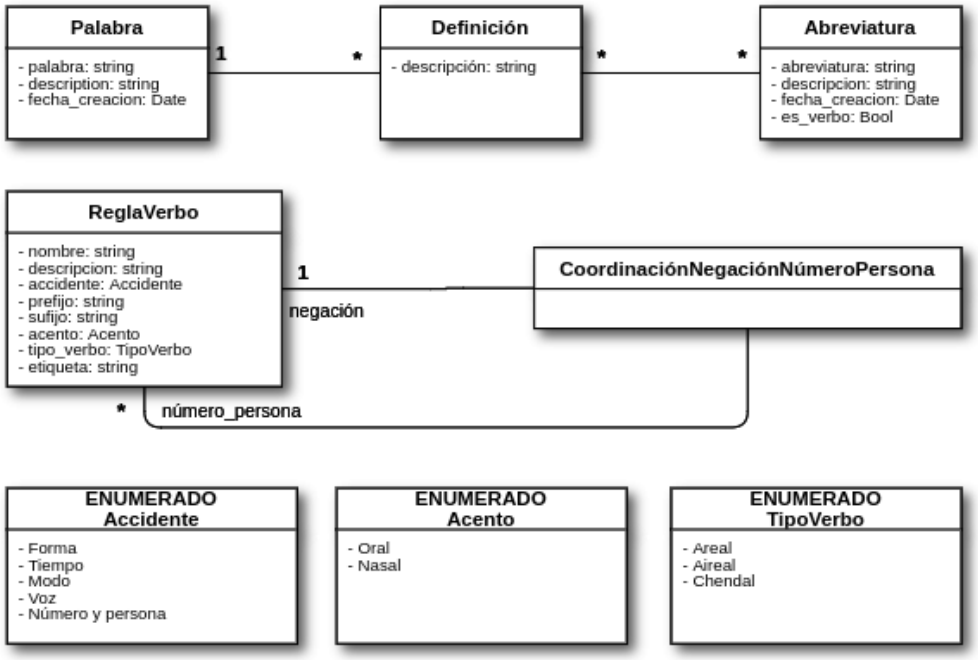

Figura 3.1: Esquema de la base de datos

En el esquema se observan los modelos asociados a la información proveniente del diccionario (“Palabra”, “Definición” y “Abreviatura”), el modelo “ReglasVerbo” utilizado para guardar la información de cada regla gramatical y el modelo “CoordinacionNegacionNumeroPersona” que se utiliza para registrar restricciones adicionales entre el accidente de negación y el accidente de número y persona. Por ejemplo, el accidente de negación con prefijo “nd” implica que la regla para accidente de número y persona que se aplique sea 1ra del singular o 3ra del plural/singular esto se representa con una instancia del modelo “CoordinacionNegacionNumeroPersona” relacionado a instancias de “ReglasVerbos”. En la figura 3.2 se muestra una instancia de la regla para el accidente de número y persona “1ra persona, Singular” con prefijo “a”, una instancia de la regla para el accidente de número y persona “3ra persona, Singular o Plural” con prefijo “o”, una instancia de la regla asociada al accidente verbal de negación con prefijo “nd” y la instancia de “CoordinacionNegacionNumeroPersona” que relaciona estas reglas.

Figura 3.2: Esquema de representación de "CoordinacionNegacionNumeroPersona"

Dado que el diccionario scrapeado tiene toda la información almacenada como un único string fue necesario distinguir cada una de las posibles definiciones de cada palabra junto con sus abreviaturas correspondientes para guardar esta información de manera estructurada. Por ejemplo, la palabra "mbarete" tiene como definición "adj. Fuerte, poderoso, potente, prepotente, pujante, intenso, sólido, duro, firme, vigoroso, forzudo, recio. 2. s. Fuerza, potencia, ímpetu, vigor, prepotencia, violencia. 3. adv. Sólidamente, fuertemente.". Esta información se estructura en la base como una palabra y tres definiciones. Cada una de estas definiciones a su vez están asociadas a diferentes abreviaturas como se muestra en la figura 3.3.

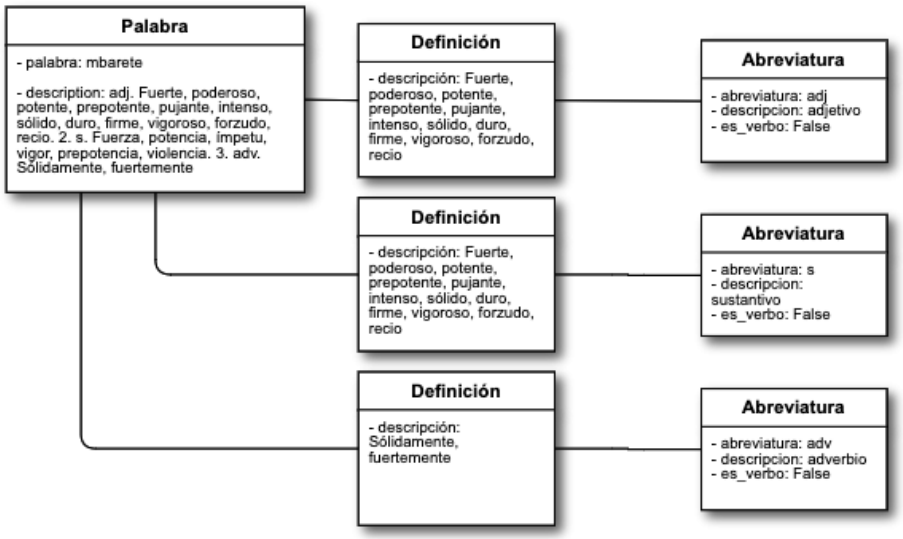

Figura 3.3: Esquema de representación de la palabra "mbarete"

3.3. Análisis morfológico de verbos

Para el análisis morfológico de verbos se utilizó un enfoque basado en reglas. Para implementar este método primero fue necesario realizar una investigación en profundidad del idioma y de la gramática. En la sección 2.1 se detallan los aspectos relativos a la gramática del guaraní y los accidentes que se aplican a los verbos. Estos accidentes son los que definen las reglas que se implementaron para realizar el análisis de los verbos. Como se mostró en la sección 2.1 la aparición de los diferentes accidentes ocurre en un determinado orden. Esta característica del idioma fue utilizada al momento de implementar las reglas. Para realizar el análisis de accidentes, se fue reconociendo prefijos y sufijos en el orden posible de aparición, tendiendo en cuenta las restricciones impuestas por reglas como la negación, donde la aparición de un prefijo además de aportar el accidente de forma condiciona la concordancia con el número y persona del verbo.

Las palabras que contienen vocales nasales y tildes fueron consideradas de manera particular al realizar el análisis. Para el caso de los tildes, se consideró las variantes de la palabra con y sin tilde para realizar el análisis. Las vocales nasales, tienen la particularidad de que se pueden escribir de dos formas, utilizando la virgulilla de la eñe (˜) o el acento circunflejo (ˆ) indistintamente. Se acuerda arbitrariamente utilizar vocales nasales con la virgulilla de la eñe. Todas las palabras que tuvieran acento circunflejo en el diccionario fueron reemplazadas por su versión con la virgulilla de la eñe. Por otro lado, cada vez que se va a realizar el análisis de una palabra, previamente se corrigen las vocales nasales que no estén en la forma acordada.

El resultante que queda luego de sacar los prefijos y sufijos reconocidos para cada combinación posible se denomina lema. Es la raíz del verbo y lo que le da el significado.

De este modo se obtienen todas las posibles combinaciones de prefijos, sufijos y lemas que se encuentran para una palabra dada basándonos en las reglas gramaticales del idioma.

Cada una de las combinaciones posibles se representa como un string que contiene todas las etiquetas correspondientes a los accidentes con-

catenadas y además el lema del verbo. En caso de que algún accidente no aplique se le agrega una etiqueta especial para indicar esto. Por ejemplo, para indicar que no se aplica accidentes de forma se agrega la etiqueta "INDEFFORM". Además el orden en el que aparecen las etiquetas es siempre el mismo (tiempo, forma, número y persona, voz, modo, lema). De esta forma todas las secuencias de etiquetas generadas para las palabras tienen largo 6. Uno de los verbos que aparece en el corpus es "omoheñoiva'ekue" (que en español significa originado), el mismo se transforma en el string "TPRETPLUSCUAMPERFECTO++INDEFFORM++3SINPLU++VACTSIMPLE++MINDSIMPLE++moheñoi". Esto es una concatenación de los accidentes de tiempo ("TPRETPLUSCUAMPERFECTO": pretérito pluscuamperfecto), forma ("INDEFFORM": no presenta accidente de forma), número y persona ("3SINPLU": tercera persona del singular o plural), voz ("VACTSIMPLE": voz activa simple), modo ("MINDSIMPLE": modo indicativo simple) y el lema resultante ("moheñoi"). Por lo tanto, la representación de la palabra "omoheñoiva'ekue" en el corpus es ("TPRETPLUSCUAMPERFECTO++INDEFFORM++3SINPLU++VACTSIMPLE++MINDSIMPLE++moheñoi", "V").

Para decidir cuál de todas las posibles combinaciones se corresponde con el análisis correcto se deben priorizar las mismas. Para esto, luego de tener todas las posibles combinaciones de prefijos y sufijos aplicables a una palabra, se analiza el lema restante utilizando el diccionario construido previamente 3.2. Las opciones son priorizadas según las condiciones que aparecen en el cuadro 3.2.

Categoría	Descripción
1	El lema resultante o la palabra se encuentra en el diccionario y tiene una definición asociada a una abreviatura que es verbo.
2	El lema resultante se encuentra en el diccionario y su abreviatura no es un verbo, pero se aplicó alguna regla gramatical de los accidentes verbales.
3	El lema resultante se encuentra en el diccionario y su abreviatura no es un verbo, tampoco se aplicó alguna regla gramatical de los accidentes verbales.
4	El lema resultante no se encuentra en el diccionario, pero se aplicó alguna regla gramatical de los accidentes verbales.
5	El lema resultante no se encuentra en el diccionario, tampoco se aplicó alguna regla gramatical de los accidentes verbales.

Cuadro 3.2: Priorización de Análisis Morfológicos

Dentro de cada categoría se prioriza según la cantidad de reglas aplicadas, cuántas más reglas se apliquen mayor es la prioridad. En caso de ser la misma cantidad se utiliza el orden de aparición en el árbol de resultados devuelto por el algoritmo.

Para contabilizar la aplicación de reglas no se tienen en cuenta las que no requieren prefijos ni sufijos. Por ejemplo, la regla para indicar voz activa simple no adiciona morfemas al lema, por lo que al no encontrar prefijos o sufijos que indiquen un accidente de voz diferente se asume que la voz expresada por el verbo es la activa simple. Esta regla no es contabilizada para el criterio establecido anteriormente.

A modo de ejemplo veamos los posibles análisis para la palabra “oikókuri” (vivieron/vivió). En las figuras 3.4, 3.5, 3.6 y 3.7 se muestran algunos de los análisis posibles para la palabra.

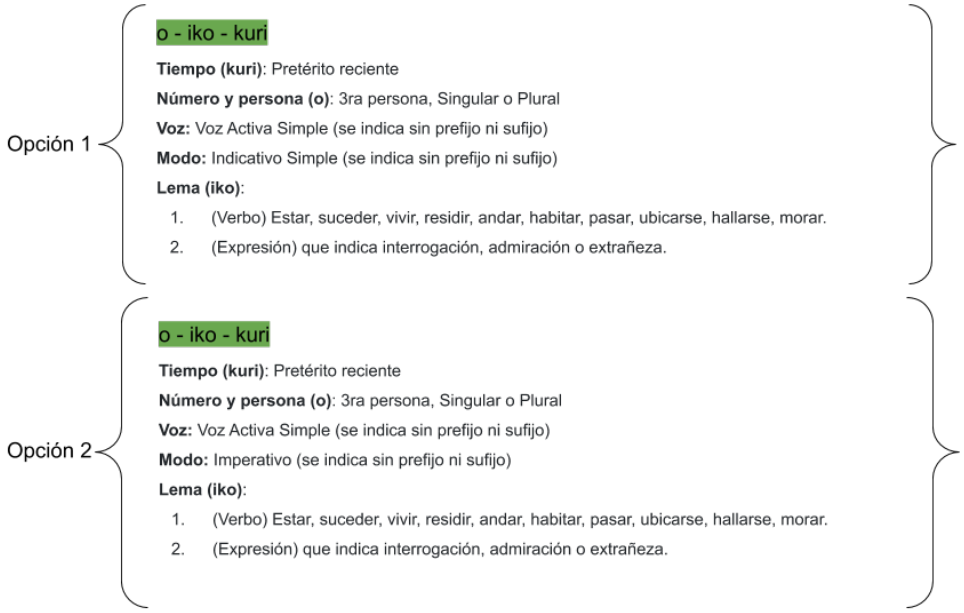

Figura 3.4: Análisis: “oikókuri” (vivieron/vivió)

En la figura 3.4 vemos las opciones que corresponden a la categoría 1 en la priorización 3.2. Estas son combinaciones en las cuales el lema resultante “iko” se encuentra en el diccionario y además tienen una definición asociada como verbo. La opción 1 es el análisis correcto para este ejemplo dado que el tiempo (pretérito reciente), el número y persona (3ra del plural/singular), la voz (activa simple) y el modo (indicativo simple) son las conjugaciones verbales correctas para el verbo “vivieron/vivió”, además de que el lema “iko” indica el significado correcto (“vivir”).

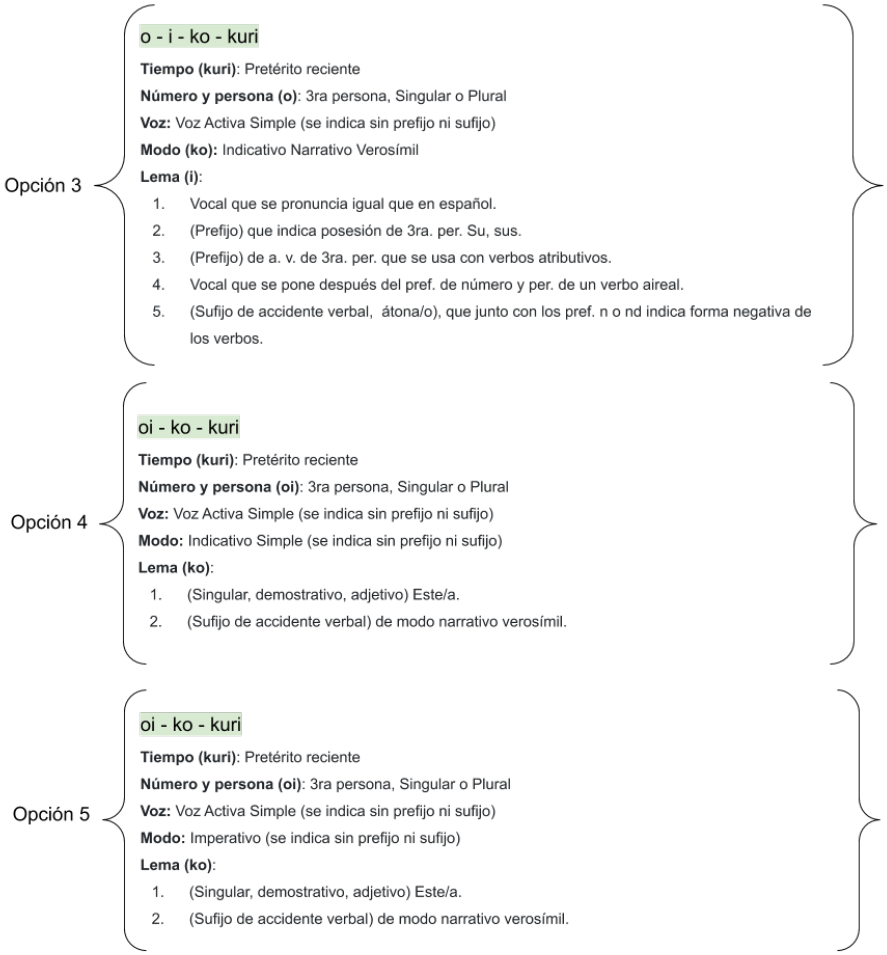

Figura 3.5: Análisis: “oikókuri” (vivieron/vivió)

En la figura 3.5 vemos otras opciones de análisis que se corresponden con la categoría 2 en la priorización 3.2, en estos casos los lemas “i” (en la opción 3) y “ko” (en las opciones 4 y 5) no se corresponden con verbos pero sí fueron encontrados en el diccionario, además se aplicó alguna regla de accidentes al encontrar los prefijos. Si bien no es tan claro que la palabra se corresponda con un verbo en estos casos, podría llegar a ser un análisis si fuera el caso de una categoría léxica verbalizada, por este motivo se le da más prioridad que a otras opciones.

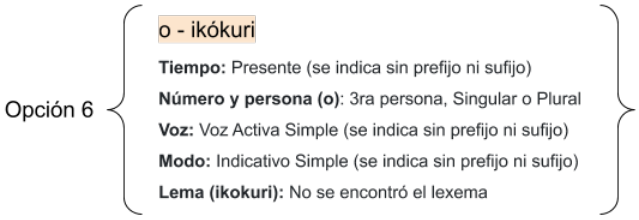

Figura 3.6: Análisis: “oikókuri” (vivieron/vivió)

En la figura 3.6 vemos que la opción que se muestra se corresponde con la categoría 4 de la priorización 3.2, el lema “ikókuri” no se encuentra en el diccionario pero el prefijo “o” podría indicar un accidente de número y persona, pero dado que no se encuentra el lema en la base se descarta este análisis de los correctos.

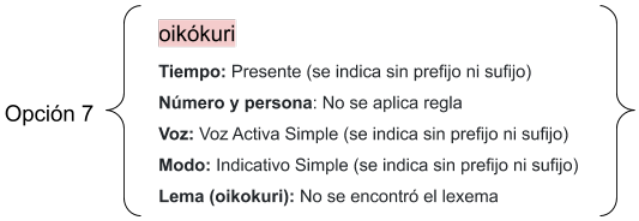

Figura 3.7: Análisis: “oikókuri” (vivieron/vivió)

En la figura 3.7 vemos un análisis correspondiente con la categoría 5 en 3.2, en el mismo se considera el lema como la palabra entera la cual no se encuentra en el diccionario y tampoco se aplican reglas contabilizables lo que nos lleva a descartar también este análisis.

En conclusión usando el esquema de prioridades definido, el analizador morfológico de verbos devuelve el análisis correcto para este verbo “oikókuri”, que es el que aparece en la figura 3.4.

Se debió realizar una consideración especial sobre algunas palabras comunes, con más de una definición que generaban ruido en el análisis. Por ejemplo, la palabra “ha” que tienen varias definiciones puede ser el sustantivo “racimo”, el verbo irregular “ir”, o la conjunción “y” entre otras. La

conjunción “y” es muy utilizada en ambos idiomas (español y guaraní), pero como también puede ser considerada en guaraní como el verbo “ir” cada vez que aparece en el corpus se interpreta como un verbo, cuando la mayoría de las veces realmente refiere a la conjunción “y”. Por esto se tomó la decisión de excluir la definición de “ha” como verbo exclusivamente, sólo se considera un verbo cuando es el lema resultado de la aplicación de alguna regla. Con criterios similares se excluye un conjunto acotado de palabras.

De esta forma el análisis con mayor prioridad es el que se considera correcto.

3.4. Detección de verbos

La detección de verbos se abordó desde dos enfoques, un método basado en reglas y un método estadístico basado en Hidden Markov Model.

3.4.1. Enfoque basado en reglas

Para el enfoque basado en reglas se elaboró una heurística para determinar cuáles palabras son verbos. Como se explicó en la sección 3.3 con el análisis morfológico se obtiene un conjunto priorizado de posibles combinaciones de prefijos y sufijos en los cuales es posible descomponer la palabra. Para saber si la palabra es un verbo o no se estudian todas las posibles combinaciones encontradas en el análisis anterior, si al menos una es candidata se puede afirmar que la palabra posiblemente sea un verbo. Decimos que una combinación es candidata si cumple al menos una de las siguientes condiciones:

1. Si el lema resultante o la palabra se encuentra en el diccionario y tiene una definición asociada a una abreviatura que es un verbo, entonces se considera que la palabra es un verbo. Se corresponde con los análisis con prioridad 1 en 3.2.
2. Si el lema resultante se encuentra en el diccionario y su abreviatura no es un verbo, pero se aplicó alguna regla gramatical de los accidentes verbales, se considera que la palabra es un verbo. Este es el caso de las categorías léxicas verbalizadas. Se corresponde con los análisis con prioridad 2 en 3.2.

Una vez establecido el análisis para una palabra, se procede a realizar el mismo análisis sobre cada una de las palabras de una oración, permitiendo detectar los verbos existentes en la misma y su correspondiente análisis de accidentes.

Este método se fue ajustando utilizando el conjunto de entrenamiento y posteriormente el conjunto de desarrollo. Luego se realizó la evaluación con

el conjunto de test.

3.4.2. Enfoque estadístico basado en HMM

El segundo enfoque utilizado para detectar verbos es un método estadístico basado en Hidden Markov Model. Para la implementación de este método se utilizaron las librerías nltk [7] y sklearn [23].

HMM con palabras originales

El modelo es entrenado con el submódulo Hidden Markov Models [31] del módulo Tagger Interface de la librería nltk. Este submódulo recibe como entrada los datos del conjunto de entrenamiento estructurados de la siguiente forma:

$$\begin{bmatrix} [(\text{“}pal11\text{”}, \text{“}NV\text{”}), (\text{“}pal12\text{”}, \text{“}V\text{”}), (\text{“}pal13\text{”}, \text{“}V\text{”}), ...], \\ [(\text{“}pal21\text{”}, \text{“}V\text{”}), (\text{“}pal22\text{”}, \text{“}V\text{”}), (\text{“}pal23\text{”}, \text{“}NV\text{”}), ...] \\ .... \\ [(\text{“}palN1\text{”}, \text{“}V\text{”}), (\text{“}palN2\text{”}, \text{“}NV\text{”}), (\text{“}palN3\text{”}, \text{“}V\text{”}), ...] \end{bmatrix}$$

Es un array que contiene oraciones y cada una es representada como un array de palabras. Las palabras a su vez son tuplas que contienen la palabra y la etiqueta “V” (si la palabra es un verbo) o “NV” (si no es un verbo).

HMM con lema original

Luego se realizó el mismo entrenamiento pero esta vez en lugar de utilizar las palabras como estaban en las oraciones del corpus, se transformaron las palabras en una concatenación de etiquetas que representan sus accidentes verbales más una etiqueta con el lema. Para esto se aplicó a cada palabra el análisis morfológico implementado con el método basado en reglas 3.3.

HMM con lema etiquetado

Finalmente se realizó el mismo experimento pero utilizando una variante del análisis morfológico en la cual se concatena el resultado de los accidentes y una etiqueta que representa si el lema resultante es o no un verbo según el diccionario en lugar de utilizar el lema original ("LEMAESVERBO" - "LEMANOESVERBO"). Por ejemplo, para la palabra "omoheñoiva'ekue" se concatena la etiqueta "LEMAESVERBO" dado que el lema resultante (moheñoi) es un verbo según el diccionario utilizado. Resultando en la representación ("TPRETPLUSCUAMPERFECTO++INDEFFORM++3SINPLU++VACTSIMPLE++MINDSIMPLE++LEMAESVERBO", "V").

Este formato de representación ayuda a reducir el problema de dispersión de datos. Este fenómeno se da cuando hay muchos datos diferentes que aparecen muy pocas veces en el corpus provocando que las probabilidades obtenidas tiendan a cero.

Estos dos últimos experimentos pueden considerarse un modelo híbrido entre el método basado en reglas y el modelo basado en Hidden Markov Model.

Todos los experimentos para este modelo se realizaron entrenando los modelos con el conjunto de entrenamiento. Luego se evaluaron estas pruebas utilizando los conjuntos de desarrollo y test.

3.5. Herramienta web

Se desarrolló una herramienta web con una interfaz de usuario sencilla que permite consultar palabras en el diccionario, realizar el análisis morfológico de verbos y detectar verbos en una oración. Esta herramienta fue desarrollada en Pyhton Django 3.0 y se utilizó la base de datos relacional en PostgreSQL [24] implementada previamente 3.2. Tanto el análisis como la detección de verbos fueron realizados en base al método basado en reglas implementado.

La herramienta tiene dos funcionalidades básicas, una permite consultar una palabra y realizar el análisis morfológico de la misma, y la otra es detectar los verbos presentes en una oración. En la interfaz se muestran dos opciones "Analizar verbo" y "Analizar Oración" que se corresponde con las funcionalidades anteriores.

El análisis morfológico se realiza seleccionando la opción "Analizar Verbo". Como se muestra en la figura 3.8, el usuario debe ingresar una palabra y el sistema realizará el análisis del verbo mostrando todos los análisis posibles, indicando con diferentes colores la probabilidad de que el análisis sea correcto y efectivamente la palabra sea un verbo en base a la priorización establecida en la tabla 3.2.

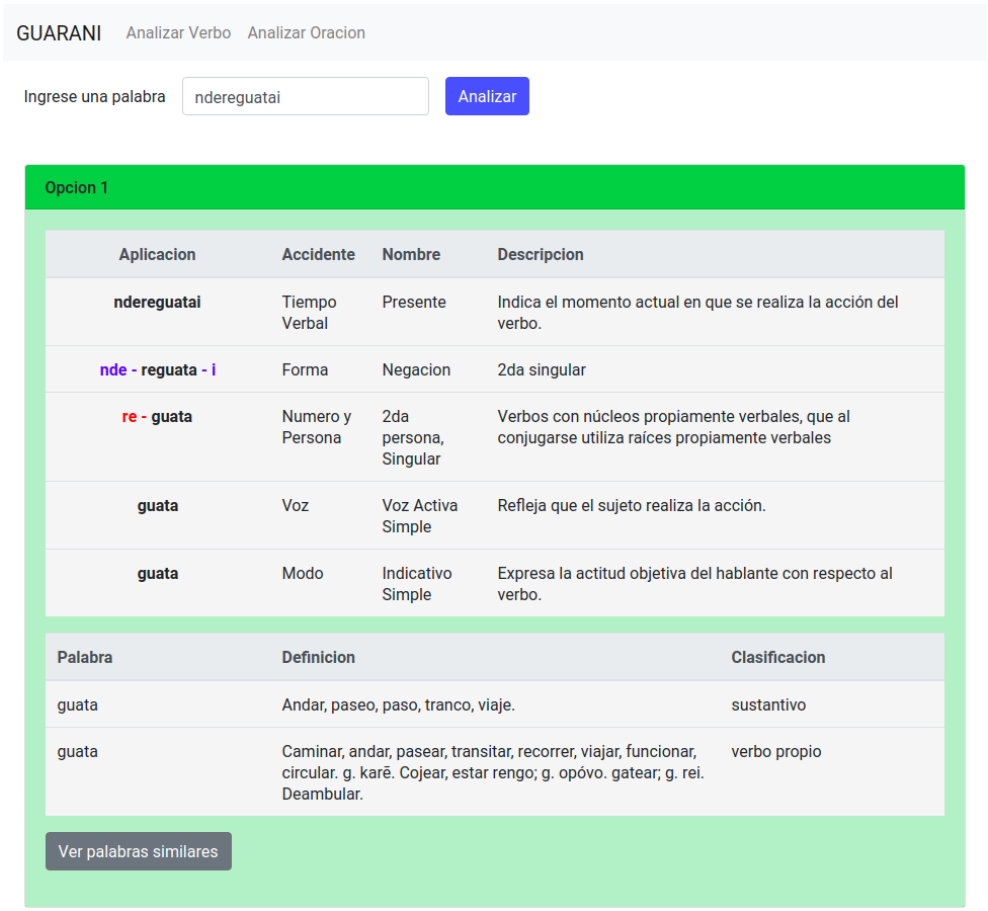

Figura 3.8: Análisis: “ndereguatai” (no caminas)

Para este ejemplo se puede ver el caso de la negación del verbo “guata” (caminar) conjugado en segunda persona del singular y en tiempo presente. La herramienta devuelve varios análisis, en la figura se muestra la primera opción. Esta se muestra en color verde porque el lema resultante de la aplicación de reglas es un verbo por lo que se puede asegurar que la palabra que se está analizando también lo es.

En contraposición en la figura 3.9 se intenta analizar la palabra “mandi” que es un adverbio que significa “enseguida o inmediatamente”, el resultado se marca en amarillo dado que el análisis probablemente sea incorrecto. Esto es porque no se aplicó ninguna regla gramatical para verbos, y el lema resul-

tante no es un verbo, sin embargo no se marca en rojo porque el lema si fue encontrado en la base. De todas formas, se descarta que sea un verbo.

Figura 3.9: Análisis: “mandi” (‘enseguida o inmediatamente)

En la figura 3.10 se muestra el resultado de realizar el análisis de la palabra “omoheñoiva’ekue”. Para este ejemplo se realiza una explicación más completa mostrando varios análisis que devuelve la herramienta para la palabra.

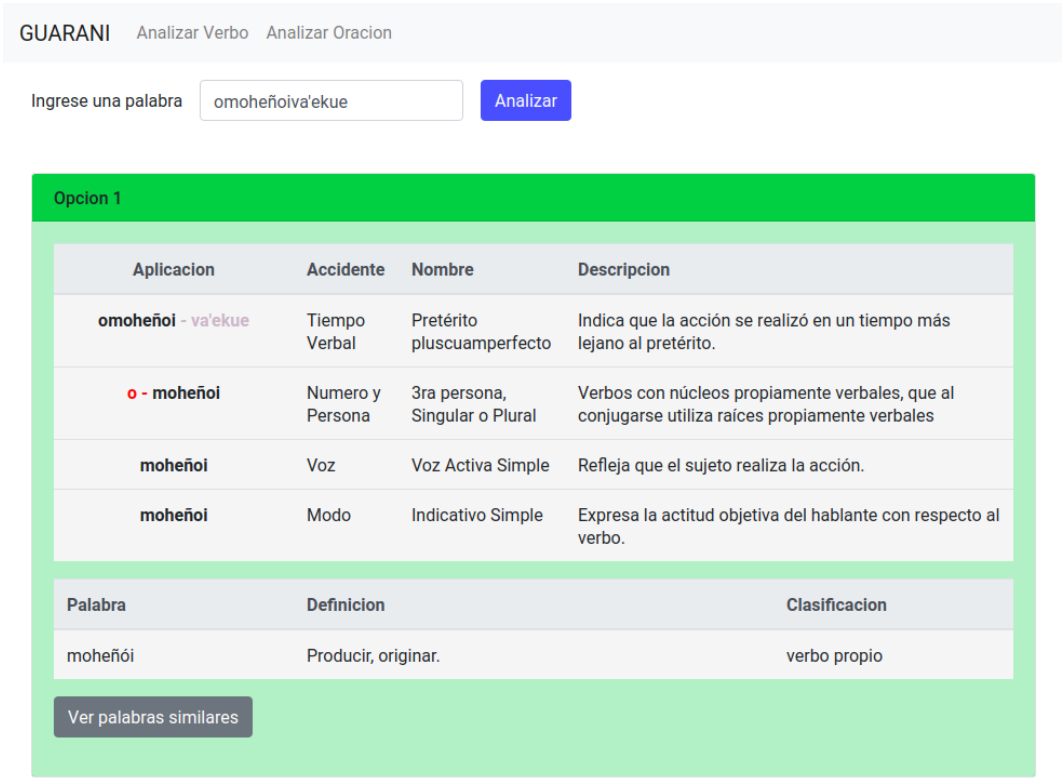

Figura 3.10: Análisis: “omoheñoiva’ekue” (produce)

Primero se muestran en color verde las opciones que cumplen con los criterios establecidos en la heurística 3.4.1 definida para la detección de verbos con el método basado en reglas, esta se corresponde con las categorías 1 y 2 de la priorización 3.2. En la figura 3.11 se muestra el segundo análisis que realiza la herramienta, este también se indica con color verde ya que es un posible análisis correcto al igual que la primera opción. Estos dos casos pertenecen a la categoría 1 de la priorización 3.2, en caso de haber opciones en la categoría 2 se mostrarían en un tono de verde más claro.

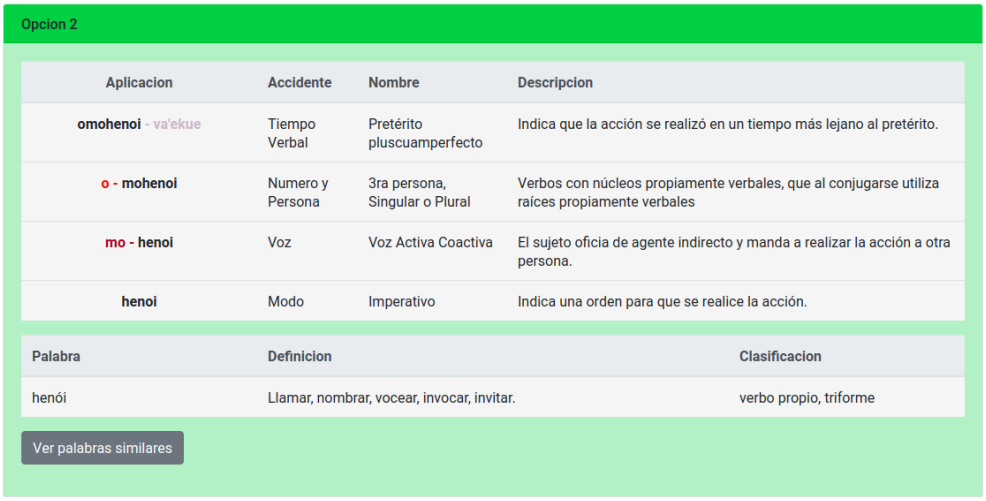

Figura 3.11: Opción 2 para análisis de palabra: “omoheñoiva’ekue”

Luego, se muestran las combinaciones en las que el lema se encuentra en la base pero no se corresponden con un verbo, y no se aplican reglas para verbos (en color amarillo). Para el ejemplo no existe un análisis que cumpla con lo anterior. Con menor probabilidad se encuentran las combinaciones para las cuales no se encontró el lema en la base pero se aplicó alguna regla para verbos (en color naranja). En el caso del ejemplo se corresponde con el análisis mostrado en la figura 3.12.

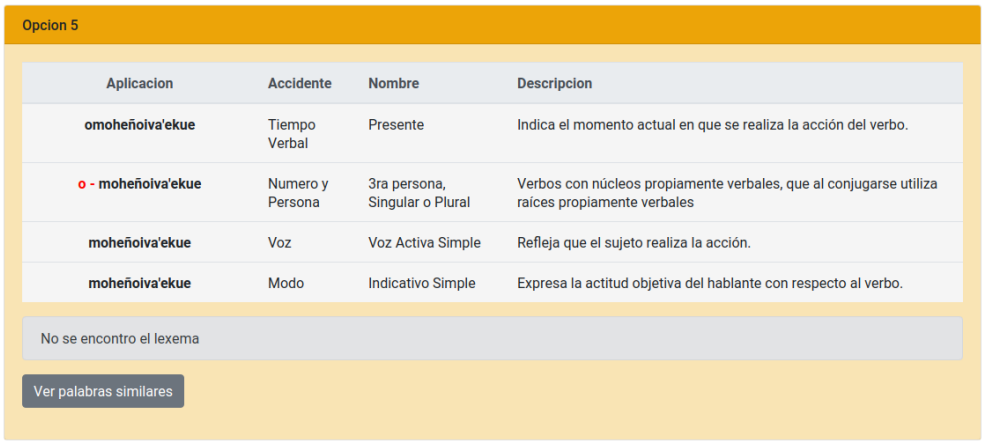

Figura 3.12: Opción 5 para análisis de palabra: “omoheñoiva’ekue”

Y por último se muestran las combinaciones en las que no se encontró el lexema en la base ni se aplicaron reglas para verbos (en color rojo). En el ejemplo, una de las opciones que se corresponde con este caso es la que se muestra en la figura 3.13

Figura 3.13: Opción 11 para análisis de palabra: “omoheñoiva’ekue”

En caso de que la palabra ingresada coincida exactamente con alguna definición del diccionario se muestran estas definiciones en un cuadro informativo previo al análisis. Por ejemplo, en la figura 3.14 al analizar la palabra “guata” se puede ver previo al análisis las definiciones de la palabra en la base.

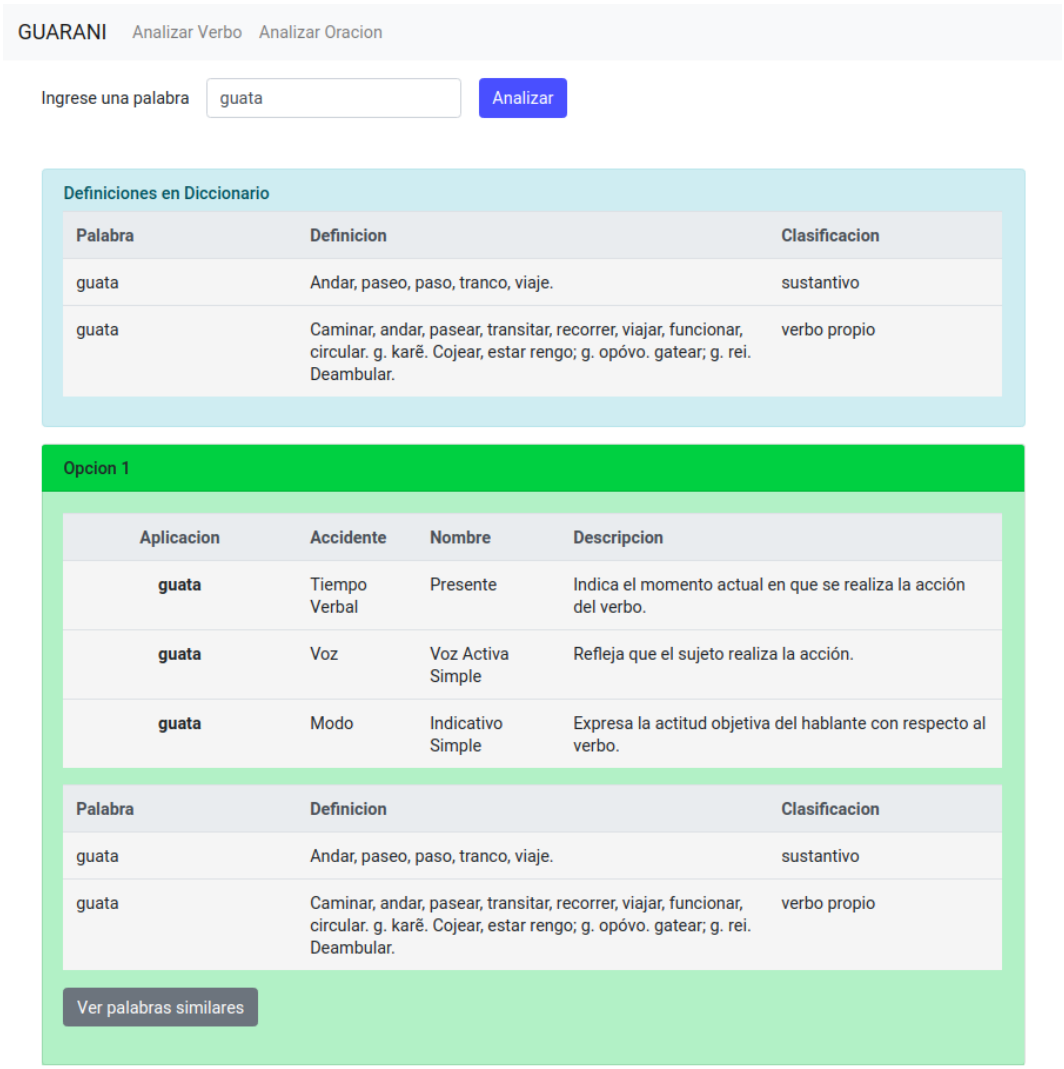

Figura 3.14: Análisis: “guata”

Para los lemas que se obtienen de cada posible análisis se muestran las definiciones que haya en el diccionario para el lema exacto, y la posibilidad de ver las palabras que contengan en ese lexema. Por ejemplo, en la figura 3.14 se ve la opción para desplegar las palabras similares.

La detección de verbos se realiza mediante la opción “Analizar Oración”. El usuario debe ingresar la oración y el sistema desplegará como resultado todas las palabras que hayan sido clasificadas como verbos, con la opción de ver sus posibles análisis de manera individual. En la imagen 3.15 se muestra el análisis de la oración “Nohendúi vaicha chupe ha upéramõ hatãve osapukái”, que en español significa “No parecía escucharle y entonces gritó más fuerte”. Se observa que el resultado del análisis devuelve las palabras “nohendúi” (no escucha) y “osapukái”(gritó) que son los verbos que tiene la oración. Cada una de las palabras es un desplegable que le permite ver los posibles análisis de estas palabras, de manera similar a como se visualizan en la opción de análisis por palabras individuales.

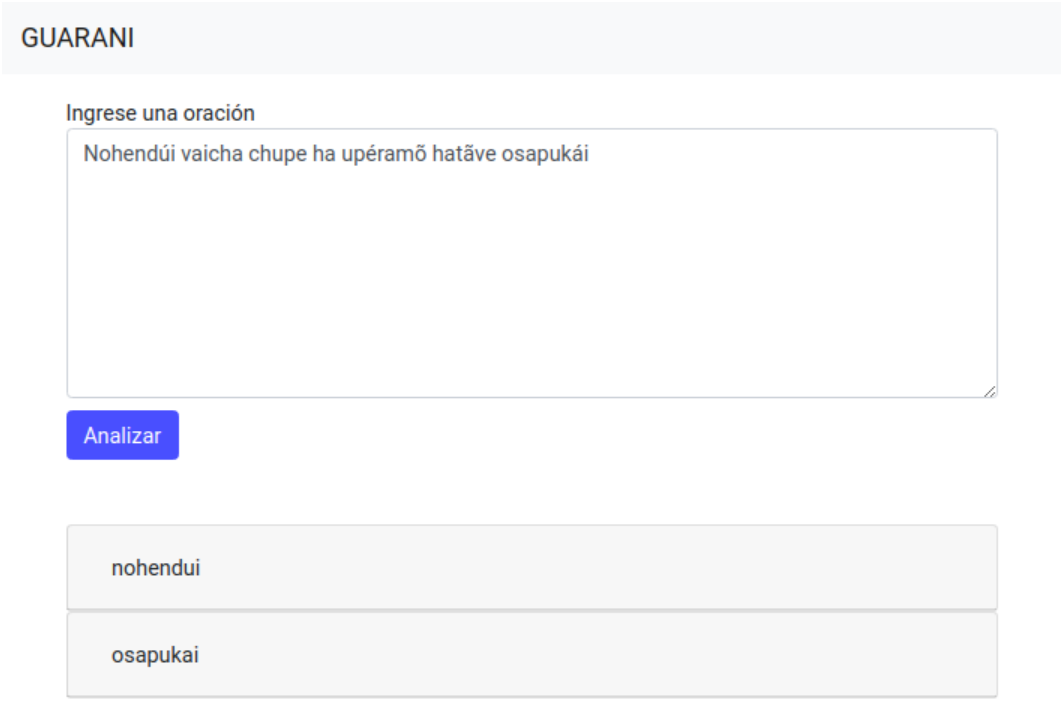

Figura 3.15: Análisis: “Nohendúi vaicha chupe ha upéramõ hatãve osapukái”

3.5.1. Ambiente de trabajo

El repositorio[2] se encuentra publicado en GitLab. El proyecto fue desarrollado en Ubuntu 18.04.4. Para la instalación se debe tener PostgreSQL [24] instalado previamente, y contar con un usuario "postgres". Se implementó un script en Shell que se ejecuta con el comando "./start_db.sh", el mismo instala todas las dependencias necesarias en el proyecto, incluyendo "python3", "Django 3.2.0", "nltk", "scikit-learn" entre otras librerías necesarias. Una vez realizada la instalación, se debe levantar el servidor, para esto se debe ejecutar el comando "python3 manage.py runserver". El proyecto queda accesible en "http://localhost:8000/".

[2]https://gitlab.fing.edu.uy/proyectogrado2019/analizadorguarani.git

3.6. Traducción Automática

El problema de traducción automática se aborda con un método estadístico basado en redes neuronales. Se implementó el método utilizando la librería OpenNMT [16]. Este método no fue estudiado en detalle, sólo se utilizó la librería a modo de "caja negra" para tener un estimativo de qué tan bien funciona un método basado en redes neuronales para la traducción del idioma con el corpus dado.

OpenNMT utiliza tres conjuntos denominados *train*, *test* y *validation*. Dado que se precisa un corpus de gran tamaño se utilizó el corpus original completamente, este incluye noticias, cuentos y blogs.

El conjunto de *train* se formó con archivos del conjunto de entrenamiento definido en la sección 3.1 y otros archivos de noticias. De esta manera se utilizaron 10,069 oraciones para este conjunto (78 % del corpus).

El conjunto de *test* se formó con los archivos del conjunto de desarrollo definidos en la sección 3.1 y otro archivos de noticias. En total cuenta con 1,469 oraciones (11.5 % del corpus).

El conjunto de *validation* se formó sólo con archivos de noticias. Este conjunto tiene 1,342 oraciones (10.5 % del corpus).

En una primera instancia se corrió el programa para tener una noción inicial. Se utilizaron los parámetros que trae OpenNMT definidos por defecto que son 100,000 iteraciones y puntos de validaciones cada 5,000, este experimento demoró alrededor de tres días. Con la configuración expresada anteriormente se corrió una vez más el programa. Esta vez cada palabra que es clasificada como verbo por el método basado en reglas se transformó en etiquetas consecutivas que representan los accidentes verbales que componen al verbo y al final se concatenó el lema del mismo. Tanto la detección de los verbos como su análisis morfológico fueron realizados en base al método de reglas. Por ejemplo, la palabra "omoheñoiva'ekue" que es un verbo, se transformó en las palabras "TPRETPLUSCUAMPERFECTO INDEFFORM 3SINPLU VACTSIMPLE MINDSIMPLE moheñoi".

Los resultados de los experimentos anteriores reflejaron que los mejores

modelos se generan en las primeras iteraciones, más precisamente para iteraciones menores a la 20,000. Por este motivo se decidió reducir el número de iteraciones a 20,000 con puntos de validación cada 2,000 iteraciones y realizar diferentes pruebas sobre los datos. Utilizando el método basado en reglas se transformó cada palabra que califica como verbo en diferentes representaciones. A continuación se enumeran los experimentos realizados con esta configuración.

1. **Original:** Primero se realizó el experimento sin realizar cambios en los archivos, sólo modificando la cantidad de iteraciones. Por ejemplo, al verbo "omoheñoiva'ekue" no se le realiza ningún cambio.

2. **Separación en accidentes verbales:** Luego se convirtió cada verbo en etiquetas consecutivas que describen sus accidentes, y al final se agrega el lema resultante. Por ejemplo, la palabra "omoheñoiva'ekue" que es un verbo, se transformó en las palabras "TPRETPLUSCUAMPERFECTO INDEFFORM 3SINPLU VACTSIMPLE MINDSIMPLE moheñoi".

3. **Separación en accidentes verbales sin etiquetas por defecto:** El tercer experimento es muy similar al anterior pero excluyendo las etiquetas de accidentes que no implican agregar prefijos o sufijos al lema, es decir los accidentes que se atribuyen por defecto cuando no aplica ninguna de las reglas. También se decidió sacar la etiqueta que indica el accidente de tercera persona ya que en la mayoría de los casos se utiliza esta forma. Por ejemplo, la palabra "omoheñoiva'ekue", se transformó en las palabras "TPRETPLUSCUAMPERFECTO moheñoi".

4. **Etiquetado con lema original:** El cuarto experimento consiste en utilizar como representación de cada verbo una concatenación de sus accidentes mediante el símbolo "++" y al final se agrega el lema. Por ejemplo, la palabra "omoheñoiva'ekue" que es un verbo, se transformó en la palabra "TPRETPLUSCUAMPERFECTO++INDEFFORM++3SINPLU++VACTSIMPLE++MINDSIMPLE ++ moheñoi".

5. **Etiquetado con lema etiqueta:** El quinto experimento es similar al anterior con la diferencia que en lugar de concatenar al final el lema

del verbo, se agrega una etiqueta que indica si el lema es o no un verbo en la base de datos. Por ejemplo, la palabra "omoheñoiva'ekue" que es un verbo, se transformó en la palabra "TPRETPLUSCUAMPERFECTO++INDEFFORM++3SINPLU++VACTSIMPLE++MINDSIMPLE++LEMAESVERBO". Con esta representación podrían llegar a haber diferentes palabras que se correspondan con la misma representación.

Cabe destacar que los experimentos 2, 3, 4 y 5 pueden ser considerados modelos híbridos entre el método basado en reglas y el método basado en redes neuronales.

En cada caso la conversión se aplica para los archivos de los conjuntos de *train*, *test* y *validation* en su versión en guaraní ya que son utilizados como input del algoritmo. Luego de hacer la predicción de los archivos de test se evalúa el resultado.

Los valores obtenidos para la nueva configuración mostraron que para la iteración 12,000 se alcanzaban los mejores resultados. Por esta razón se hizo una segunda evaluación de los experimentos con el conjunto de test definido en la sección 3.1, es decir fue utilizado como nuevo conjunto de *test* para OpenNMT. Esta última prueba se realiza con el objetivo de observar si los resultados se mantienen al utilizar un conjunto nuevo, que no fue usado para entrenar, ni validar, ni testear y por lo tanto es desconocido para el sistema.

Capítulo 4

Resultados

En esta sección se evaluarán las soluciones implementadas para el análisis morfológico de verbos, la detección de verbos y la traducción automática.

4.1. Análisis morfológico de verbos

Este problema fue abordado mediante la implementación de un método basado en las reglas gramaticales del idioma guaraní. Para evaluarlo se definieron las siguientes métricas:

- Exact accuracy: corresponde a una medida estricta y se calcula como la suma de los aciertos en el análisis de verbos, divido la cantidad total de verbos en el corpus. Es decir, se compara que el análisis realizado por el método sea exactamente igual al esperado.

  Sea n la cantidad total de palabras en el corpus y x_i con $i \in \{1, ..., n\}$ las palabras del corpus. Se define la función $F(x_i) = y_i$ con $i \in \{1, ..., n\}$ donde $y_i = 1$ si la clasificación de la palabra x_i es correcta, $y_i = 0$ en otro caso.

$$exact\_accuracy = \frac{y_1 + y_2 + ... + y_n}{cantidad\_total\_de\_verbos} \tag{4.1}$$

- Relaxed accuracy: es una medida más relajada que la anterior. Utilizando el etiquetado de las palabras es posible definir esta métrica que permite evaluar el accuracy de las secuencias de etiquetas encontradas. Para calcular esta medida se considera para cada palabra el promedio de aciertos de la secuencia, es decir la cantidad de etiquetas acertadas dividido la cantidad de etiquetas en la secuencia. Luego se hace el promedio general como la suma de los promedios por palabra dividido la cantidad de verbos en el corpus.

  El largo de la secuencia de etiquetas es fijo y siempre vale 6 cualquiera sea la palabra. Fue posible realizar esta asunción por el formato en que se implementó el etiquetado de las palabras, en el cual cada accidente verbal se corresponde con una etiqueta aunque el valor tenga que ser no aplica o indefinido.

  Sea n la cantidad total de palabras en el corpus y x_i con $i \in \{1, ..., n\}$ las palabras del corpus. Se define $F(x_i, j) = e_{ij}$ con $i \in \{1, ..., n\}$ y $j \in \{1, ..., 6\}$ el resultado de la clasificación de la etiqueta j de la palabra i del corpus. Para la etiqueta e_{i6} dependiendo del formato que se esta evaluando corresponde al lema original o a la etiqueta que determina si el lema es o no un verbo en el diccionario. Si la etiqueta j fue clasificada correctamente para la palabra i entonces $e_{ij} = 1$, en otro caso $e_{ij} = 0$.

$$relaxed\_accuracy = \frac{\frac{e_{11}+e_{12}+e_{13}+e_{14}+e_{15}+e_{16}}{6} + ... + \frac{e_{n1}+e_{n2}+e_{n3}+e_{n4}+e_{n5}+e_{n6}}{6}}{cantidad\_total\_de\_verbos} \tag{4.2}$$

Estas métricas fueron calculadas sobre los conjuntos de desarrollo y test, en la figura 4.1 se puede observar los valores obtenidos. La representación "Lema original" refiere a la explicada en la sección 3.3, consiste en la concatenación de las etiquetas que representan los accidentes verbales más el lema resultante. La representación "Lema etiquetado" refiere a la explicada en la sección 3.4.2 bajo el nombre "HMM lema etiquetado". Es similar a la representación anterior pero reemplazando el lema por una etiqueta que indica si el lema es un verbo o no en la base.

Métricas		Conjunto	
		Desarrollo	Test
Exact accuracy	Lema original	0.436	0.310
	Lema etiquetado	0.386	0.304
Relaxed accuracy	Lema original	0.751	0.615
	Lema etiquetado	0.745	0.621

Cuadro 4.1: Resultados de accuracy para el método basado en reglas

En la figura 4.1 y 4.2 se muestran los valores de exact accuracy y relaxed accuracy obtenidos.

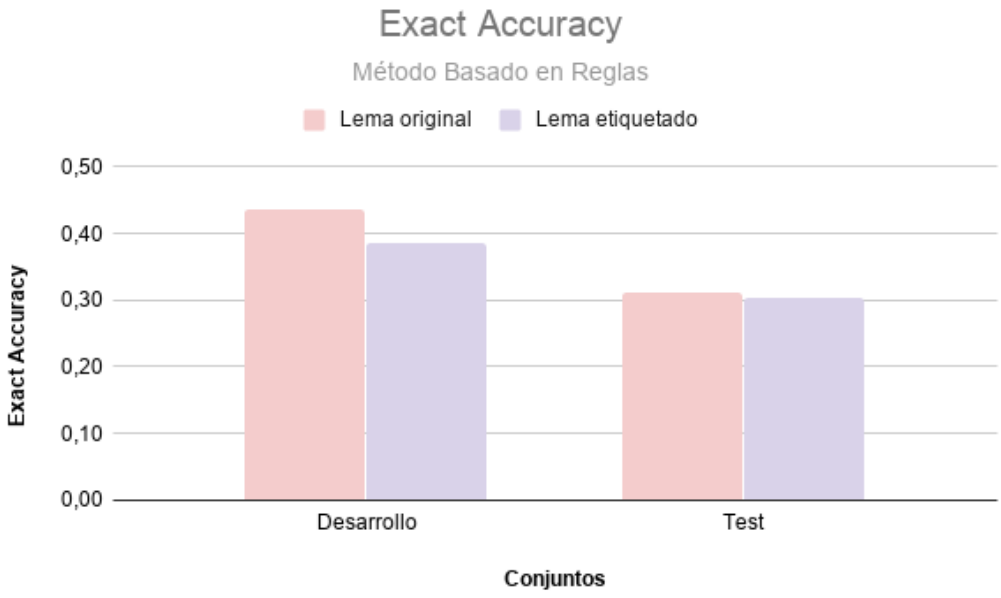

Figura 4.1: Exact Accuracy

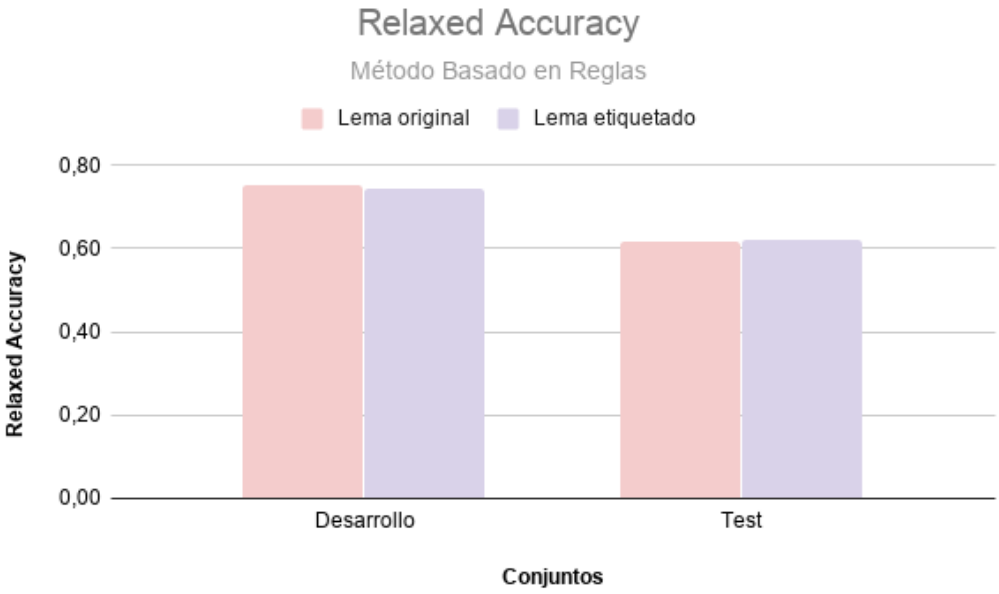

Figura 4.2: Relaxed Accuracy

Se puede observar que la medida de relaxed accuracy en cada caso dio más alta que la de exact accuracy. Esto significa que algunas palabras no son etiquetadas de la manera exacta en que se espera, sin embargo varios de sus accidentes son encontrados correctamente. El exact accuracy para ambos experimentos (lema original y lema etiquetado) dan resultados en el entorno de 0.3 para el conjunto de test, por lo que utilizar estos métodos para la clasificación morfológica de verbos como única herramienta no sería apropiado. Sin embargo, el relaxed accuracy devuelve valores superiores a 0.6 en todos los casos, lo que indica que en muchos casos si bien el análisis morfológico no es exactamente igual al esperado, varios accidentes verbales de la palabra son etiquetados correctamente.

Se pude concluir que este método permite tener una buena aproximación al análisis morfológico de los verbos.

4.2. Detección de verbos

Para evaluar las soluciones implementadas para la detección de verbos es necesario contabilizar la cantidad de aciertos que se obtienen con cada uno de ellos. Los aciertos son determinados en función del análisis manual que se realizó previamente, se considera que se acertó una clasificación cuando la clasificación manual coincide con la clasificación que se obtiene con el método utilizado. Para evaluar estos métodos se utilizaron las medidas Precision, Recall, FScore y Accuracy detalladas a continuación.

- **Precision:** refiere a la probabilidad de que una observación sea realmente positiva cuando el clasificador la etiqueta como positiva. En otras palabras, de las instancias que se etiquetaron como positivas cuántas realmente lo eran. Se calcula como la relación entre los verdaderos positivos (TP) y la suma de los verdaderos positivos (TP) más los falsos positivos (FP):

$$\frac{TP}{TP + FP} \tag{4.3}$$

  TP: instancias clasificadas correctamente como positivas

  FP: instancias clasificadas como positivas pero que en realidad son negativas

- **Recall:** se define como la probabilidad de identificar las instancias positivas como tales. En otras palabras, de todas las instancias que son positivas cuántas encuentra el clasificador. Se calcula como la relación entre los verdaderos de positivos (TP) y la suma de los verdaderos positivos (TP) más los falsos negativos (FN):

$$\frac{TP}{TP + FN} \tag{4.4}$$

  TP: instancias clasificadas correctamente como positivas

  FN: instancias clasificadas como negativas pero que en realidad son positivas

- **F1Score:** se define como la media harmónica entre precision y recall:

$$2 * \frac{precision * recall}{precision + recall} \tag{4.5}$$

Si aumenta mucho la precision disminuye el recall, lo que significa que cuando el clasificador devuelve positivo tiende a acertar el resultado, pero clasifica pocas instancias como positivas. Si por el contrario aumenta demasiado el recall, el clasificador devuelve más instancias positivas pero comienza a bajar la precision, es decir comienza a generar más falsos positivos.

- **Accuracy:** Es el promedio de aciertos que tiene el método, se calcula como la relación entre la cantidad de verbos acertados por el método y el total de verbos.

  [27]

A continuación se presentan los resultados de Precision, Recall, F1Score y Accuracy obtenidos para los distintos métodos de identificación de verbos.

Métricas		Conjunto	
		Desarrollo	Test
Precision	Método basado en reglas	0.574	0.611
	HMM con palabras originales	0.975	0.872
	HMM con lema original	0.975	0.891
	HMM con lema etiquetado	0.822	0.859
Recall	Método basado en reglas	0.838	0.716
	HMM con palabras originales	0.191	0.069
	HMM con lema original	0.191	0.083
	HMM con lema etiquetado	0.546	0.510
Fscore	Método basado en reglas	0.681	0.660
	HMM con palabras originales	0.319	0.128
	HMM con lema original	0.319	0.153
	HMM con lema etiquetado	0.656	0.640
Accuracy	Métodos basado en reglas	0.874	0.862
	HMM con palabras originales	0.871	0.871
	HMM con lema original	0.871	0.871
	HMM con lema etiquetado	0.909	0.909

Cuadro 4.2: Resultados detección de verbos

En las figuras 4.3, 4.4, 4.5 y 4.6 se muestra la comparación de estas métricas utilizando los diferentes métodos implementados. Sólo se graficaron las métricas sobre las clasificaciones positivas, es decir sobre las palabras que se corresponden con verbos. Las métricas sobre las instancias clasificadas como no-verbo son la mayoría en el corpus y no aportan información relevante a la hora de comparar los métodos.

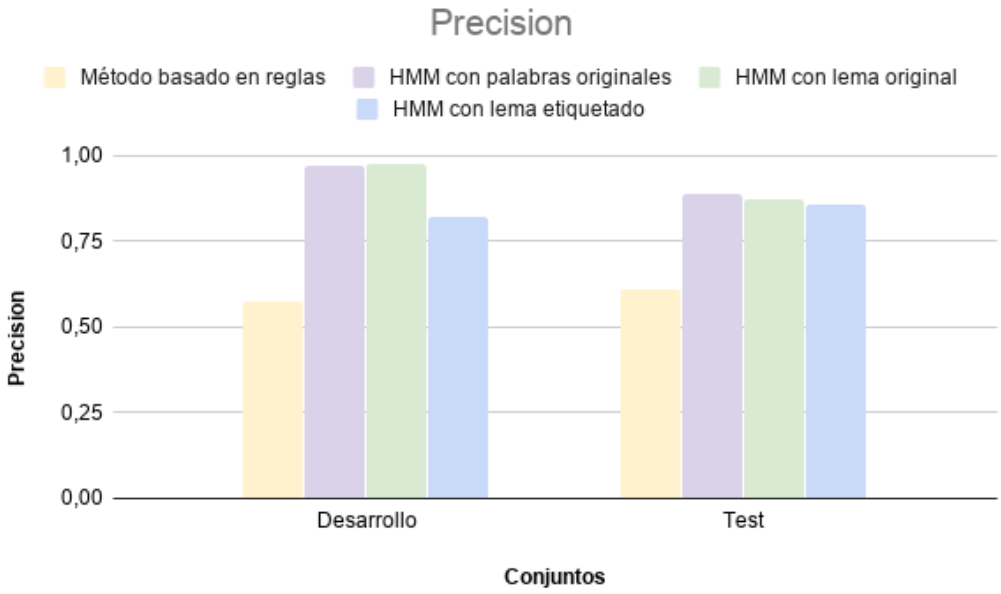

Figura 4.3: Precision

De la gráfica 4.3 se deduce que el método basado en Hidden Markov Model con lema original y con palabras originales son los que devuelven los mejores valores de presicion. Esto indica que estos métodos tienden a acertar el resultado cuando clasifican una palabra como verbo. De todas formas los otros métodos se encuentran por encima de 0.5, lo cual es un buen resultado.

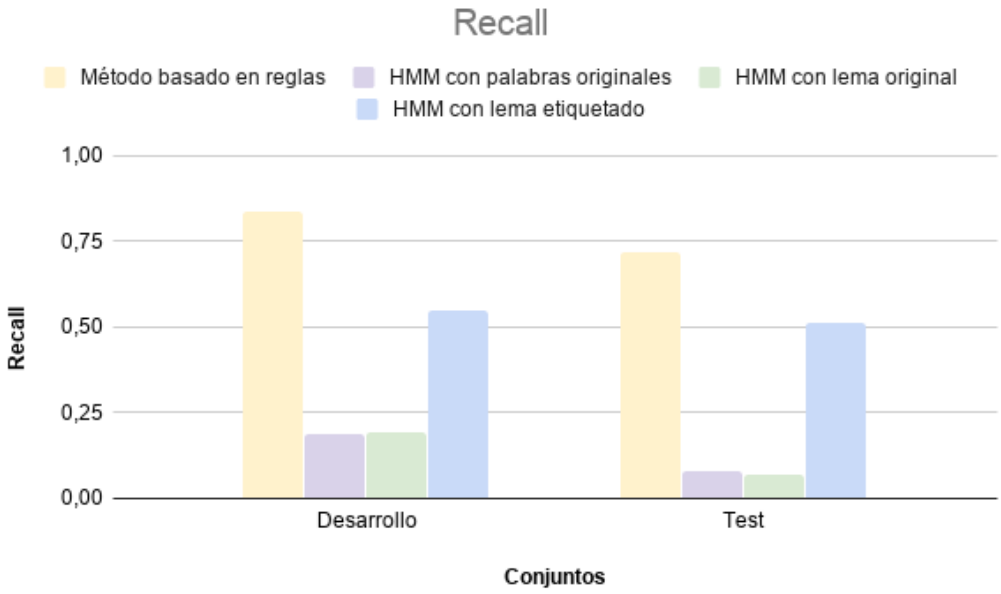

Figura 4.4: Recall

En cuánto al recall se observa que el método basado en reglas es notoriamente mayor a los demás métodos. Esto indica que tiende a clasificar mayor cantidad de instancias como verbos, mientras que los métodos basados en Hidden Markov Model tienden a clasificar menos. Al observar el resultado de recall para el método basado en Hidden Markov Model con lema original y con palabras originales, se puede ver que los valores son mucho menores a los demás, sin embargo con estos métodos se obtuvieron la mayor precision, esto nos indica que es un método que tiende a acertar cuando clasifica una palabra como verbo, pero dado su recall se deduce que tiende a clasificar muy pocas palabras como verbos.

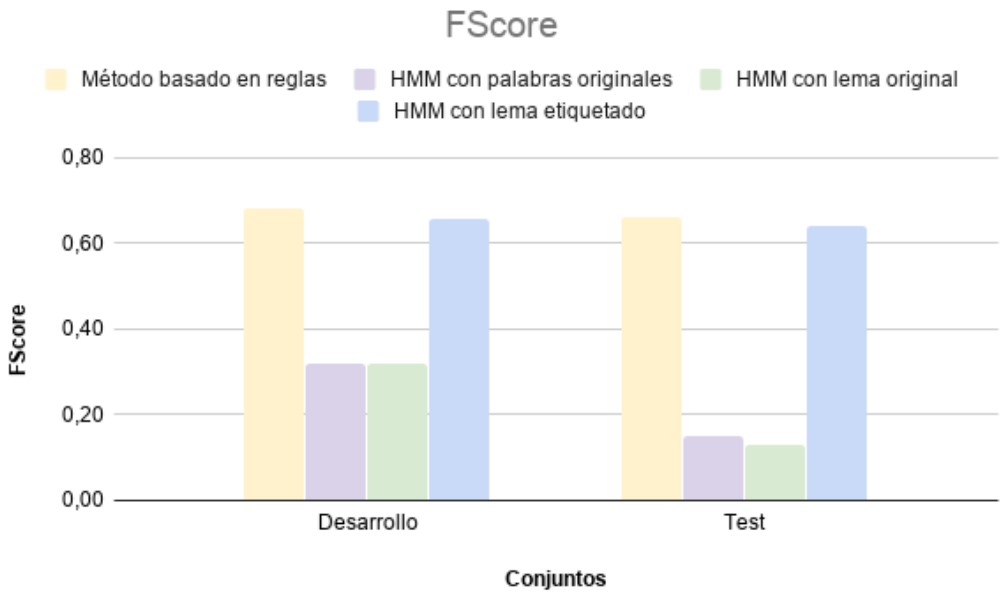

Figura 4.5: FScore

La medida FScore refleja la relación entre las medidas anteriores (recall y precision). Se observa que los valores para el método basado en reglas y el basado en Hidden Markov Model con lema etiquetado son similares y en el entorno de 0.60. Sin embargo para los métodos basado en HMM con lema original y con palabras originales da un resultado visiblemente menor.

Al evaluar los resultados de los métodos con mejor FScore, se observa que con el método basado en reglas se obtiene mayor recall, mientras que con el método basado en Hidden Markov Model con lema etiquetado se obtiene mayor precision. Estas diferencias se complementan generando el FScore similar. Ambos métodos muestran ser adecuados para la clasificación de verbos. La diferencia es que el método basado en Hidden Markov Model con lema etiquetado tiene mayor precision cuando clasifica una palabra como verbo, pero etiqueta pocos verbos. Por otro lado, el método basado en reglas tiene menor precision al momento de clasificar una palabra como verbo pero clasifica más instancias como verbos.

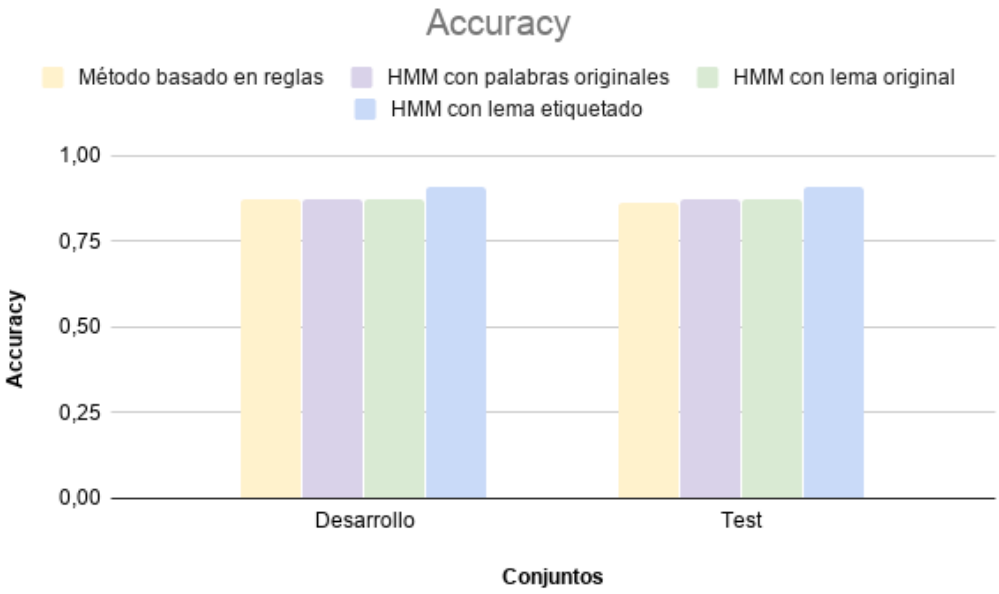

Figura 4.6: Accuracy

En cuánto al accuracy se puede ver que los valores son muy similares para todos los métodos. Para ambos conjuntos el mejor resultado lo tiene el método basado en HMM con lema etiquetado. Sin embargo, esta medida está influenciada por el desbalance de las clases, hay una mayor cantidad de no-verbos que de verbos. Por esta razón se considera que esta medida no es tan significativa para este estudio, a diferencia de la medida FScore que permite evaluar los aciertos sobre una clase específica.

El resultado de evaluar estas métricas induce a concluir que los mejores métodos para la clasificación de verbos en el idioma guaraní dado el corpus existente son el método basado en reglas, y el híbrido basado en Hidden Markov Model con los lemas etiquetados.

4.3. Traducción Automática

Para evaluar los resultados obtenidos en traducción automática se utilizó la medida BLEU.

BLEU (Bilingual Evaluation Understudy) es una medida utilizada para evaluar la calidad de traducción de un traductor automático. La calidad de la traducción refiere a la similitud que existe entre la traducción automática y las traducciones humanas de referencia para una misma frase de origen. El algoritmo consiste en comparar expresiones consecutivas de la traducción humana con expresiones consecutivas en la traducción automática, ponderando el total de coincidencias independientemente de la posición en que aparezcan. La medida BLEU puede valer entre 0 y 1, donde 1 representa la correspondencia perfecta entre la traducción esperada y la candidata. Sin embargo esto es casi imposible de lograr ya que en el problema de traducción generalmente pueden existir varias traducciones válidas. Cuántas más coincidencias se encuentren, más alto será el BLEU, indicando mayor grado de similitud entre la traducción esperada y la traducción automática. El resultado de esta medida depende de la amplitud del dominio, la cantidad de datos disponibles para el entrenamiento y la coherencia de los datos esperados con los datos de entrenamiento y validación. Si el modelo se entrena en un dominio reducido, y los datos de aprendizaje son coherentes con los datos de prueba, se induce a que la medida BLEU sea alta.[2] Se recurrió a la función corpus_blue[29] de la librería nltk para la obtención de esta medida.

A continuación se muestra los valores obtenidos de la ejecución con la configuración por defecto tanto para la versión original de los archivos como para le versión etiquetada.

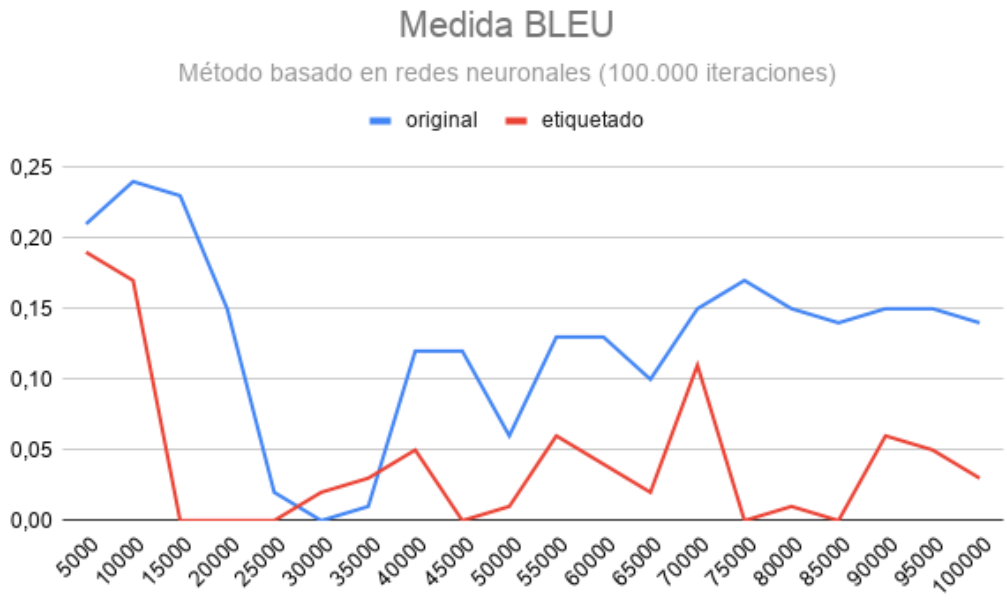

Figura 4.7: Resultados BLEU de OpenNMT para 100,000 iteraciones

Como se mencionó en la sección 3.6 los mejores resultados se obtienen en las primeras iteraciones llegando a alcanzar un BLEU de 0.237 para la versión original y 0.17 para la versión etiquetada. Como se puede observar en la gráfica 4.7 luego de la iteración 20,000 los valores obtenidos son más bajos. Se observó que el experimento con la versión original de los archivos en casi todas las iteraciones obtuvo mejores resultados frente a la versión etiquetada.

Por otro lado, además de evaluar la medida BLEU se observó manualmente las predicciones realizadas por el modelo. Estas presentan muy bajo nivel de fluidez y fidelidad lo cuál condice con el bajo valor de la medida BLEU. Por ejemplo, para la iteración 75,000 de la versión original, la oración "Dijo que en la reunión representantes del Gobierno propusieron una nueva reunión, pero no se definió fecha para ello." se traduce en "Dijo que la COMPRA en la COMPRA de la COMPRA de la COMPRA de la COMPRA de la COMPRA". Se observa que la traducción no es semánticamente correcta (baja fidelidad), y tampoco es fluida ya que "la COMPRA" se repite varias

veces careciendo de sentido.

A continuación se muestran los resultados para los siguientes experimentos que se realizaron. Los mismos fueron detallados en la sección 3.6.

Iteración	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
2000	0.106	0.205	0.179	0.230	0.204
4000	0.213	0.148	0.203	0.237	0.220
6000	0.204	0.188	0.255	0.210	0.236
8000	0.218	0.197	0.227	0.201	0.237
10000	0.222	0.233	0.235	0.244	0.229
12000	0.244	0.195	0.263	0.238	0.279
14000	0.154	0.106	0.247	0.163	0.255
16000	0.001	0.149	0.252	0.022	0.249
18000	0.077	0.053	0.227	0.047	0.060
20000	0.091	0.003	0.093	0.135	0.147

Cuadro 4.3: Resultados BLEU de OpenNMT para 20,000 iteraciones

En la gráfica 4.8 se puede observar el comportamiento de los resultados para los experimentos realizados para 20,000 iteraciones.

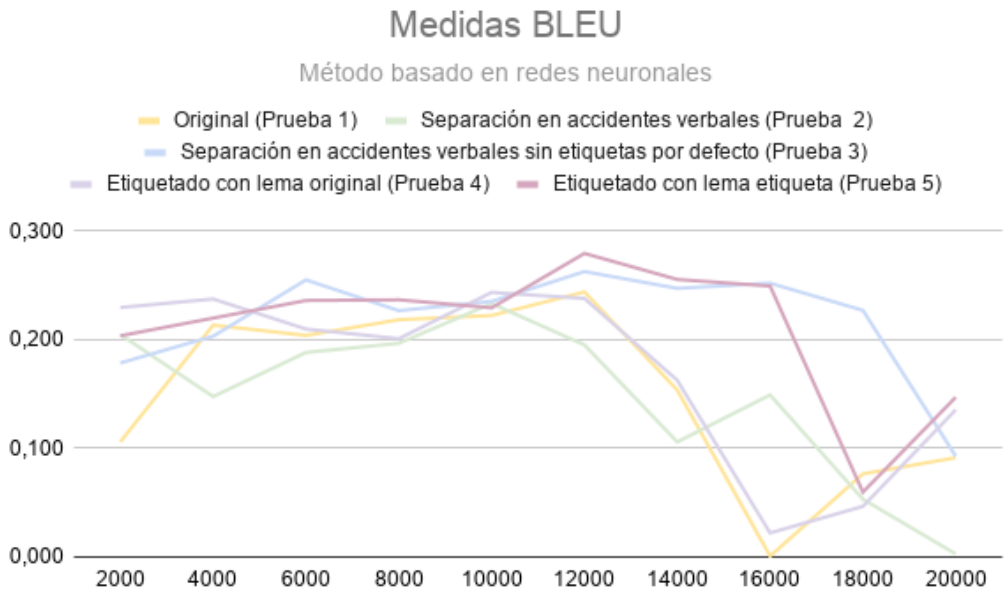

Figura 4.8: Resultados BLEU de OpenNMT para 20,000 iteraciones

Se puede apreciar que los experimentos que destacaron con mejores resultados fueron las pruebas 3 y 5, estas alcanzan un BLEU de 0.263 y 0.279 respectivamente para la iteración 12,000. Sin embargo, el experimento que dio peores resultados en la mayoría de los casos es la Prueba 2. Por otro lado, las Pruebas 1 y 4 se comportan de manera similar. Esto parece indicar que utilizar el método basado en reglas de manera complementaria mejora los resultados obtenidos.

Por último, se muestran los valores obtenidos para las pruebas anteriores sobre el conjunto de test para la iteración 12,000.

Iteración	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
12000	0.174	0.138	0.174	0.157	0.203

Cuadro 4.4: Resultados BLEU de OpenNMT para la iteración 12,000

Se observó que se mantenía el mismo comportamiento, es decir la prueba

5 con el resultado más alto, luego las prueba 3, 1 y 4 y por último la prueba 2. Por lo tanto, con un conjunto de archivos totalmente nuevo para el sistema se mantuvo la relación de los resultados obtenidos, lo cual es un indicador de que efectivamente utilizar modelos híbridos mejora los resultados respecto a la versión original.

En general, llegamos a la conclusión de que el mejor resultado se obtiene con la prueba 5 que se corresponde con el experimento que consiste en transformar cada palabra en una concatenación de accidentes verbales más el lema resultante. En contraposición, con la prueba 2 se obtienen los peores resultados, la misma se corresponde con la transformación de cada palabra en un conjunto de 6 etiquetas que describen sus 5 accidentes y el lema resultante. Creemos que este bajo resultado se debe a la representación y no al análisis morfológico realizado con el método basado en reglas. Si sacamos este experimento un poco particular la prueba con peores resultados es la que se obtiene sin hacer ninguna modificación al conjunto, por lo tanto consideramos que utilizar el método híbrido es un avance en la línea de investigación.

Para la prueba 5, que es la que mejores resultados obtuvo se volvió a hacer una revisión manual de las predicciones. En este caso se observó que en general las traducciones tienen mayor fluidez, ya no aparecen tantas repeticiones de palabras sin sentido. Por ejemplo, para la iteración 12,000, la oración "Estudiantes de natación realizan exhibición en Ayolas" se traduce en "Estudiantes verifican circuitos turísticos de Ayolas". La mayoría de las predicciones siguen teniendo errores en cuánto a su fidelidad aunque en algunos casos trasmiten mensajes parecidos.

Finalmente, dado que ningún valor de BLEU llegó a superar el valor 0.3 se concluye que, con el corpus existente, este método para la traducción automática precisa de otros métodos o herramientas en los cuales apoyarse para lograr una mejor traducción. Utilizar el método basado en reglas mejora los resultados pero siguen siendo relativamente bajos.

Capítulo 5

Conclusiones

5.1. Conclusiones

Este proyecto contribuye con investigaciones, herramientas y resultados que aportan a la traducción guaraní-español en el área de Procesamiento del Lenguaje Natural. Se logró aplicar nuevas tecnologías en la investigación del idioma guaraní, generando un punto de partida para futuras investigaciones.

Como resultado de la investigación se logró reunir una variedad de recursos lingüísticos relacionados al idioma guaraní, entre los que se destacan diccionarios, libros de gramática, corpus paralelos entre otros que fueron utilizados como base para la implementación de soluciones a los problemas planteados. A pesar de que logramos obtener varios recursos muy útiles, la información que hay es escasa. Particularmente, cuando se necesitó de un corpus de gran tamaño para aplicar algunos métodos como el basado en redes neuronales, vimos que los resultados dependen en gran medida del tamaño del corpus con el que se cuenta.

A lo largo del proyecto se lograron soluciones para los cuatros problemas planteados. Se logró implementar la base de datos que almacena estructuralmente información proveniente de un diccionario muy completo en comparación con otros que se encontraron. El mismo contiene gran parte de las

palabras utilizadas en el corpus. Esta base de datos es un recurso de fácil acceso para realizar consultas y se puede ir ampliando en la medida que se vayan encontrando o publicando diferentes recursos lo cual en consecuencia mejoraría las soluciones propuestas.

El análisis morfológico de verbos se realizó mediante un método basado en reglas el cual se basa en las reglas gramaticales del idioma guaraní investigadas.

Por otro lado se logró realizar la detección de verbos mediante dos enfoques, basado en reglas y mediante un método estadístico basado en Hidden Markov Model. También se realizaron pruebas que podrían considerarse métodos híbridos entre ambos enfoques, en las mismas utilizamos la detección y análisis de verbos basado en reglas para generar la entrada del método estadístico. Como resultado de estas pruebas vimos que utilizar este enfoque híbrido mejora los resultados en la detección de verbos.

Adicionalmente se implementó una herramienta web que permite visualizar las soluciones propuestas enfocadas en el método basado en reglas para el análisis morfológico de verbos y para la detección de los mismos. Es una herramienta muy útil sobre la cual se puede comenzar a explorar palabras del idioma. Además, tiene una interfaz intuitiva que permite al usuario tener información de manera rápida.

Finalmente se logró una primera aproximación para el problema de traducción automática con la utilización de redes neuronales. Se realizaron varios experimentos que logran mejorar la versión inicial. Varios de ellos podrían definirse como modelos híbridos entre el método basado en reglas y redes neuronales. De hecho, en estos experimentos basados en métodos híbridos fue con los que se obtuvieron mejores resultados para la traducción automática, por lo cual tenemos un indicio más de que el método basado en reglas efectivamente es un aporte para el estudio del lenguaje.

5.2. Desarrollo a futuro

Se considera que el presente trabajo se podría continuar o realizarle mejoras en los siguientes aspectos.

5.2.1. Corpus

El corpus actual presenta algunas dificultades como el tamaño reducido del mismo. Se podrían incluir textos universales como la Biblia o la Declaración Universal de los Derechos Humanos, considerando que estos textos requieren un mayor tiempo de procesamiento dada la complejidad del lenguaje que utilizan.

También se encontró que algunas oraciones no se correspondían exactamente con su traducción a pesar de transmitir el mismo mensaje. Por ejemplo, existen oraciones que difieren en el orden en que se dicen algunas cosas. En otros casos ocurre que se utilizan sinónimos en vez de la traducción exacta de la palabra por más que esta exista.

5.2.2. Método basado en reglas

Actualmente el método basado en reglas tiene un conjunto limitado de reglas gramaticales que podría extenderse y perfeccionarse incorporando reglas más específicas del idioma.

Particularmente, se podrían abarcar más reglas para los verbos como por ejemplo para los accidentes de grado que no fueron contemplados, los mismos se utilizan para señalar la intensidad de la acción. Además de esto, se podrían agregar reglas para la formación de otras entidades gramaticales relevantes para compresión del idioma como lo son los sustantivos y adjetivos.

5.2.3. Método basado redes neuronales

La traducción automática mediante este método se podría haber explorado más, por ejemplo ajustando otros parámetros que provee la librería OpenNMT. También se podría agregar nuevos experimentos con otro tipo de transformaciones sobre los archivos de entrada. Como ya se mencionó utilizar el método basado en reglas para complementar la traducción mejora los resultados por lo que se podría continuar esta línea de investigación.

A pesar de que los resultados obtenidos por este método no fueron muy buenos, se considera que es un camino que se debe continuar profundizando dado que permite independizar la calidad de la traducción del conocimiento humano sobre el idioma.

5.2.4. Recursos Lingüísticos

Actualmente el diccionario que se utiliza para el reconocimiento de verbos tiene alrededor de 6,600 palabras extraídas de [32]. Se podrían incorporar a la base otros diccionarios a modo de extender y mejorar el vocabulario existente. En esta primera versión se utilizó el diccionario guaraní-español, en iteraciones futuras se podría llegar a incluir las palabras del diccionario español-guaraní para relacionar o incluir más palabras.

5.2.5. Herramienta Web

La herramienta web desarrollada puede servir de apoyo para futuras investigaciones dado que proporciona conocimiento sobre el idioma guaraní (traducciones de palabras y reglas del idioma) de una manera sencilla y de rápido acceso. Sin embargo se podría mejorar, en primer lugar se podría agregar el procesamiento de español a guaraní ya que actualmente solamente se tiene el inverso. También se considera que sería útil publicarlo online para que personas de todas partes puedan beneficiarse de la herramienta sin tener que bajar el repositorio y realizar la instalación del ambiente.

Índice de cuadros

Bibliografía

[1] *¿Qué es la traducción automática?* URL: https://www.sdltrados.com/es/solutions/machine-translation.html (visitado 18-01-2020).

[2] *¿Qué es una puntuación BLEU? - Custom Translator - Azure Cognitive Services — Microsoft Docs*. URL: https://docs.microsoft.com/es-es/azure/cognitive-services/translator/custom-translator/what-is-bleu-score (visitado 15-01-2020).

[3] *A Simple Introduction to Natural Language Processing*. URL: https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32 (visitado 16-07-2019).

[4] *Análisis de sentimiento: qué es y para qué se usa — Brandwatch*. URL: https://www.brandwatch.com/es/blog/analisis-de-sentimiento/ (visitado 05-11-2019).

[5] María Antonia Martí Antonín y Joaquim Llisterri Boix. *Tecnologías del texto y del habla*. Google-Books-ID: hNPMEnqfc44C. Edicions Universitat Barcelona, 2004. 264 págs. ISBN: 9788447526475.

[6] *Aplicaciones del Procesamiento del Lenguaje Natural — El Huffington Post*. URL: https://www.huffingtonpost.es/instituto-de-ingenieria-del-conocimiento/aplicaciones-del-procesamiento-del-lenguaje-natural_a_23322448/ (visitado 10-11-2019).

[7] Edward Loper Bird Steven y Ewan Klein. «Natural Language Processing with Python». En: (2009).

[8] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra y Robert L. Mercer. «The Mathematics of Statistical Machine Translation: Parameter Estimation». En: *Computational Linguistics* 19.2 (1993), págs. 263-311. URL: https://www.aclweb.org/anthology/J93-2003 (visitado 20-10-2019).

[9] Luis Chiruzzo, Pedro Amarilla, Adolfo Ríos y Gustavo Giménez Lugo. «Development of a Guarani - Spanish Parallel Corpus». En: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, mayo de 2020, págs. 2629-2633. URL: https://www.aclweb.org/anthology/2020.lrec-1.320.

[10] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu y Pavel Kuksa. «Natural Language Processing (Almost) from Scratch». En: *NATURAL LANGUAGE PROCESSING* (2011). URL: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

[11] *Curso IPLN 2019*. URL: https://eva.fing.edu.uy/pluginfile.php/256856/mod_resource/content/1/INTROPLN%20-%20An%C3%A1lisis%20L%C3%A9xico%20y%20HMM.pdf (visitado 03-08-2019).

[12] James H. Martin Daniel Jurafsky. *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition*. Prentice-Hall, 2008. ISBN: 9780131873216.

[13] Carla Parra Escartín. *Evolución de la traducción automática*. URL: http://www.lalinternadeltraductor.org/n16/traduccion-automatica.html (visitado 19-01-2020).

[14] *Frequently asked questions - OpenNMT*. URL: https://opennmt.net/FAQ/#who-is-behind-opennmt (visitado 17-01-2020).

[15] Universidad de Granada. *Capítulo 10 - Cadenas de Markov*. URL: https://www.ugr.es/~bioestad/_private/cpfund10.pdf (visitado 13-08-2019).

[16] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart y Alexander M. Rush. «OpenNMT: Open-Source Toolkit for Neural Machine Translation». En: *Proc. ACL*. 2017. DOI: 10.18653/v1/P17-4012.

URL: https://www.aclweb.org/anthology/P17-4012.pdf (visitado 10-09-2019).

[17] Marta Krasan, Cynthia Audisio, Mayra Juanatey, Juan Krojz y Mariana Lila Rodríguez. *Material de consulta para el docente en contextos de diversidad lingüística*. 2017. ISBN: 978-987-4019-41-7. URL: http://publicaciones.filo.uba.ar/sites/publicaciones.filo.uba.ar/files/Material%20de%20consulta%20para%20el%20docente%20en%20contextos%20de%20diversidad%20ling%C3%BC%C3%ADstica_interactivo_0.pdf (visitado 23-02-2020).

[18] Academia de la Lengua Guaraní. *Gramática Guaraní*. 1ª Edición. Editorial Servilibro, ago. de 2018. ISBN: 9789996759666. URL: http://academiadelalenguaguarani.org.py/images/multimedia/publicaciones/GRAMATICA_GUARANI_CASTELLANO.pdf.

[19] Juan Tornero Lucas. *Machine Learning: Modelos Ocultos de Markov (HMM) y Redes Neuronales Artificiales (ANN)*. Jun. de 2017. URL: http://diposit.ub.edu/dspace/bitstream/2445/122446/2/memoria.pdf (visitado 25-03-2019).

[20] Anke Lüdeling y Merja Kytö. *Corpus Linguistics*. Google-Books-ID: _RiQVlQfbGkC. Walter de Gruyter, 10 de dic. de 2008. 797 págs. ISBN: 978-3-11-021142-9.

[21] *Neural Machine Translation: Using Open-NMT for training a translation model*. URL: https://medium.com/hackernoon/neural-machine-translation-using-open-nmt-for-training-a-translation-model-1129a3a2a2d3 (visitado 17-01-2020).

[22] Diego Ortiz, Domingo Aguilera y Elda Marecos. *Hablemos el Guaraní - curso completo en cuatro niveles para extranjeros - primer nivel*. Editora Litocolor, 1990. ISBN: 9789000001507. URL: https://acervo.socioambiental.org/sites/default/files/documents/GIL00007.pdf (visitado 10-03-2020).

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

[24] *PostgreSQL: Linux downloads (Ubuntu)*. URL: https://www.postgresql.org/download/linux/ubuntu/ (visitado 15-04-2019).

[25] *Procesamiento de Lenguaje Natural (PLN): aplicaciones y usos*. URL: https://www.dail.es/aplicaciones-del-procesamiento-del-lenguaje-natural (visitado 05-11-2019).

[26] Antonio Miranda Raya. *Big Intelligence: Nuevas capacidades big data para los sistemas de vigilancia estratégica e inteligencia competitiva*. 2015. ISBN: 9788415061618. URL: https://books.google.com.uy/books?id=pOFTDwAAQBAJ&pg=PA198&lpg=PA198&dq=preprocesamiento,+analisis+morfologico,+analisis+sintactico&source=bl&ots=XFB-AUk4cO&sig=ACfU3U3mSar_rwZSbSKKVZA5N_M7mfyNYQ&hl=es&sa=X&ved=2ahUKEwj7jP-izaLoAhWZHbkGHf9oCDEQ6AEwBXoECAkQAQ#v=onepage&q=preprocesamiento%2C%20analisis%20morfologico%2C%20analisis%20sintactico&f=false (visitado 21-07-2019).

[27] *SAS Help Center*. URL: https://documentation.sas.com/?docsetId=casml&docsetTarget=casml_boolrule_details05.htm&docsetVersion=8.5&locale=en (visitado 03-12-2019).

[28] *Scrapy*. URL: https://scrapy.org/ (visitado 15-04-2019).

[29] *Source code for nltk.translate.bleu_score*. URL: https://www.nltk.org/_modules/nltk/translate/bleu_score.html (visitado 10-05-2019).

[30] Universidad de Stanford. *An Introduction to Hidden Markov Models*. URL: http://ai.stanford.edu/~pabbeel/depth_qual/Rabiner_Juang_hmms.pdf (visitado 28-11-2019).

[31] «Tagger Interface package, Hidden Markov Models». En: (2020). URL: http://www.nltk.org/api/nltk.tag.html?highlight=hmm.

[32] Gabriel Enrique del Valle. *Descubrir Corrientes - GUARANI*. URL: http://descubrircorrientes.com.ar/2012/index.php/diccionario-guarani (visitado 20-03-2019).

[33] *What is Machine Translation? Rule Based Machine Translation vs. Statistical Machine Translation*. URL: https://www.systransoft.com/systran/translation-technology/what-is-machine-translation/ (visitado 19-01-2020).

Apéndice A

Anexo I: Hidden Markov Model

HMM es un modelo estadístico en donde se asume que el sistema a modelar es una cadena de Markov de parámetros desconocidos. Este modelo pretende determinar los parámetros desconocidos u ocultos a partir de los parámetros observables de la cadena.

La cadena de Markov, también conocida como proceso de Markov o modelo de Markov, es una serie de evento en donde la probabilidad de que ocurra un evento depende del evento inmediato anterior.

En términos matemáticos, es un proceso estocástico en el cual si el estado actual es X_n y los estados previos $X_1, ..., X_{n-1}$ son conocidos, entonces la probabilidad del estado futuro X_{n+1} depende del estado actual X_n y no de los estados anteriores $X_1, ..., X_{n-1}$. [15]

Entonces, se tiene que para una sucesión de estados $s_1, ..., s_{n+1}$ arbitraria

$$P(X_{n+1} = s_{n+1}|X_1 = s_1, X_2 = s_2, ..., X_n = s_n) = P(X_{n+1} = s_{n+1}|X_n = s_n)$$

Para el caso de estudio, son de interés las cadenas de Markov finitas y con probabilidades de transición estacionaria. En este tipo de cadenas se dispone de un número finito k de estados posibles $s_1, ..., s_k$ en donde en cualquier instante de tiempo la cadena se encuentra en uno de esos k estados. Luego, la probabilidad de transición es la probabilidad condicionada

$$P(X_{n+1} = s_j | X_n = s_i)$$

Una cadena de Markov posee probabilidades de transición estacionarias si para cualquier par de estados s_i y s_j existe una probabilidad de transición p_{ij} tal que

$$P(X_{n+1} = s_j | X_n = s_i) = p_{ij} \text{ para } n = 0, 1, 2, ..., k$$

Las probabilidades de transición entre los estados se suele representar mediante una matriz.

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0k} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1k} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2k} \\ p_{30} & p_{31} & p_{32} & \cdots & p_{3k} \\ \vdots & & & & \vdots \\ p_{k0} & p_{k1} & p_{k2} & \cdots & p_{kk} \end{pmatrix}$$

La matriz de transición de probabilidades para una cadena de Markov finita con probabilidades de transición estacionarias es una matriz estocástica. Esto quiere decir que se trata de una matriz cuadrada en la cual sus elementos son mayores o iguales a cero y que la suma de los elementos de cada fila es igual a uno.

Otra forma de representar este tipo de cadena es a través de un diagrama de transición. Este diagrama consta de un grafo finito y dirigido en donde cada nodo representa un estado de la cadena y los arcos indican las transiciones posibles entre los estados. Dado que es un grafo dirigido, cada uno de los arcos tiene un sentido explícito. Asociado a cada arco se puede ver la probabilidad de transición correspondiente. Por ejemplo, en la imagen A.1 se puede observar que la probabilidad de ir del estado 0 al 1 es p_{01}.

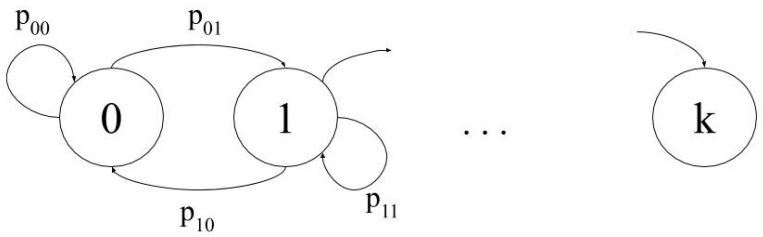

Figura A.1: Diagrama de transición de cadena de Markov

En los modelos ocultos de Markov la cadena de Markov subyace tras las observaciones. Los estados solamente pueden ser inferidos a partir de los símbolos observados. Mediante las observaciones y las transiciones de estado se busca obtener la secuencia de estados más probable.

Se dice que un HMM es un proceso doblemente estocástico. El primer proceso estocástico es un conjunto finito de estados no observable (está oculto) con probabilidades asociadas. Este puede observarse a través de un segundo conjunto de procesos estocásticos que producen la secuencia de símbolos observados. [30]

Para definir un modelo oculto de Markov se precisan cinco elementos:

1. Los N estados $S = \{S_1, ..., S_N\}$ del modelo.
2. Los M distintos símbolos observables $V = \{V_1, ..., V_M\}$. En el caso de que las observaciones sean continuas, M es infinito.
3. La matriz de transición $A = a_{ij}$, en donde $a_{ij} = P(q_{t+1} = j | q_t = i)$ y q_t es el estado actual. Se puede observar que esta matriz es equivalente a la matriz de una cadena de Markov. Se debe resaltar que si una de la probabilidad a_{ij} es cero entonces permanecerá siendo cero durante todo el proceso de entrenamiento.
4. La probabilidad de distribución de los símbolos en cada estado, siendo representada por $B = b_j(k)$ en donde $b_j(k)$ es la probabilidad de observar el símbolo v_k en el estado S_j. Esta probabilidad esta dada por

$$b_j(k) = P(o_t = v_k | q_t = j), \forall j \in \{1, ..., N\}, \forall k \in \{1, ..., M\} \qquad \text{(A.1)}$$

donde v_k representa el k-ésimo símbolo del conjunto V y o_t el vector actual de observaciones.

Las siguientes restricciones se deben cumplir

- $b_j(k) \geq 0, \forall j \in \{1, ..., N\}, \forall k \in \{1, ..., M\}$
- $\Sigma_{k=1}^{M} b_j(k) = 1, \forall j \in \{1, ..., N\}$

5. La distribución inicial de estados $\pi = \pi_i$, donde π_i es la probabilidad de que el modelo esté en el estado S_i en el tiempo inicial $t = 0$, con
   $\pi_i = P(q_1 = i), \forall i \in \{1, ..., N\}$ [19]

Una vez entendido lo anterior, se prosigue a explicar el POS tagging con HMM. Se parte de una frase, es decir de una secuencia de n palabras observables $w_1, ..., w_n$. Cada una de estas palabras puede pertenecer a una o más categorías gramaticales, estas son las etiquetas. Se quiere encontrar la secuencia n de etiquetas $t_1, ..., t_n$ mas probable, esto es el x que maximiza $f(x)$, es decir, el $argmax_x f(x)$:

$$t_1^{*n} = argmax_{t_1^n} P(t_1^n | w_1^n) \tag{A.2}$$

Luego se aplica la regla de Bayes A.3

$$P(x|y) = \frac{P(y|x).P(x)}{P(y)} \tag{A.3}$$

Esta regla proporciona probabilidades mas simples de calcular, obteniendo como resultado:

$$t_1^{*n} = argmax_{t_1^n} \frac{P(w_1^n | t_1^n).P(t_1^n)}{P(w_1^n)} \tag{A.4}$$

$P(w_1^n)$ es constante dado que no depende de t_1^n, entonces se puede realizar la siguiente simplificación:

$$t_1^{*n} = argmax_{t_1^n} P(w_1^n | t_1^n).P(t_1^n) \tag{A.5}$$

Por otro lado, se realizan dos suposiciones más en este modelo:

- La probabilidad de que una palabra ocurra depende sólo de su etiqueta (y no de otras palabras y etiquetas "alrededor")

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i) \tag{A.6}$$

- La probabilidad de que una etiqueta ocurra depende sólo de la etiqueta previa (hipótesis de bigrama).

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1}) \tag{A.7}$$

Al aplicar los resultados de las suposiciones A.6 y A.7, se llega a la siguiente igualdad:

$$t_1^{*n} = argmax_{t_1^n} P(t_1^n|w_1^n) \approx argmax_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \tag{A.8}$$

La probabilidad $P(t_i|t_{i-1})$ se calcula a partir de los ejemplos en el corpus simplemente contando:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \tag{A.9}$$

Se aplicará POS tagging con HMM a modo de ejemplo a la siguiente frase [11]

Secretariat	is	expected	to	race	tomorrow
NNP	BEZ	VBN	TO	VB	NR
				NN	

Las etiquetas utilizadas son las del corpus PennTreebank en donde significan lo siguiente:

- NNP: nombre propio, singular
- BEZ: is (verbo ser conjugado en tercera persona, singular, presente)
- VBN: verbo, participio pasado
- TO: to (auxiliar gramatical)
- VB: verbo, forma base
- NN: nombre
- NR: adverbio

Como se puede observar para esta frase hay dos posibles secuencias de etiquetas gramaticales. Esto se debe a que la palabra "race" es ambigua, puede referirse tanto al sustantivo carrera como al verbo correr.

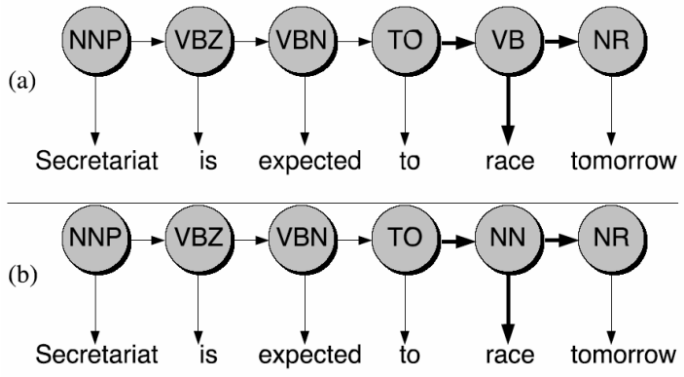

Figura A.2: Diagrama de estados [11]

Se cuenta con los siguientes datos extraídos del corpus Brown:

- $P(VB|TO) = 0{,}83$
- $P(race|NN) = 0{,}00057$

- $P(race|VB) = 0{,}00012$
- $P(NR|VB) = 0{,}0027$
- $P(NR|NN) = 0{,}0012$

A continuación se realizan los cálculos para la parte de la cadena que afecta al resultado final de la probabilidad total.

Para la opción a) en donde "race" se considera verbo se obtiene

$P(VB|TO).P(NR|VB).P(race|VB) = 0{,}00000027$

Para la opción b) en donde "race" se considera nombre se obtiene

$P(NN|TO).P(NR|NN).P(race|NN) = 0{,}00000000032$

Finalmente se puede concluir que la secuencia más probable es

Secretariat	is	expected	to	race	tomorrow
NNP	BEZ	VBN	TO	VB	NR

la cual es la que tiene más sentido.