Índice general

Etiquetador automático de la morfología del otomí usando predicción estructurada

Diego Alberto Barriga Martínez

8 de mayo de 2020

1 Introducción

1.1. Lengua otomí

En esta sección se menciona los lugares donde se describe el idioma otomí de forma somera, se mencionan algunos lugares donde es hablado el otomí y características fundamentales de la lengua.

1.1.1. Origen

La palabra otomí es de origen náhuatl (singular: *otomitl*, plural: *otomí*). Por otra parte, los otomíes se nombran a sí mismos *ñähñu*[1], que significa "los que hablan otomí".

Los grupos indígenas que hablan el idioma otomí se encuentran en diversas partes del territorio mexicano como: Estado de México, Querétaro, Hidalgo, Puebla y Veracruz (Barrientos López, 2004). El otomí es una lengua indígena una gran variación dialectal que depende de su distribución geográfica.

En el Estado de México el pueblo *ñähñu* está disperso por varios municipios tales como: Toluca, Lerma, Chapa de Mota, Aculco, Amanalco, Atizapán de Zaragoza, por mencionar algunos. En otros municipios como Naucalpan, Ecatepec, Nezahualcóyotl y Tlalnepantla se pueden encontrar hablantes por efectos de la migración. Según Barrientos López (2004) la población total de hablantes otomíes en el Estado de México supera los cien mil, sin embargo, datos actuales .

[1]Existen organizaciones indígenas, como el Consejo de la Nacionalidad Otomí, que escriben la auto-denominación como hñätho hñähñu y también ñätho ñähño. Sin embargo, esta auto denominación puede variar.

En concreto existen **nueve** variantes del otomí y cabe recalcar que dicha variación puede presentarse incluso dentro del mismo estado. Tan solo el Estado de México presenta tres variantes del otomí: El otomí de Tilapa, hablado en el municipio de Santiago Tianguistenco; el Otomí de Acazulco, del municipio de San Jerónimo Acazulco; y el Otomí de Toluca, de San Andrés Cuexcontitlán.

1.2. Problemática

El NLP es un área de la computación que permite reconocer, procesar e interpretar el lenguaje humano dentro de un sistema computacional. El objetivo de esta área es hacer que las computadoras realicen tareas que involucran el lenguaje humano. Algunas tareas generales consisten en permitir la comunicación humano-máquina, o simplemente hacer un exitoso procesamiento de texto o voz humanos (**?**). Ejemplos de aplicación actuales son los traductores automáticos, asistentes personales que reconocen voz, motores de búsquedas en Internet, análisis de sentimientos en textos, síntesis de voz, etiquetado de textos y muchas más aplicaciones.

Una de las tareas más populares de NLP es el etiquetado automático de textos. Este etiquetado puede realizarse a diferentes niveles lingüísticos, por ejemplo, morfosintáctico (*Part Of Speech tagging, POS*), sintáctico (*parsing*), morfológico, etc. El nivel morfológico tiene que ver con la estructura interna de las palabras (**?**); en particular, existe un tipo de etiquetado, de gran importancia para el análisis lingüístico, llamado glosado que asigna etiquetas a las unidades que conforman a una palabra.

Para lograr lo anterior, los enfoques actuales aplican técnicas de ML. El ML es un subcampo de la Inteligencia Artificial (IA), que constituye un enfoque de resolución de problemas caracterizado por estimar una solución a partir de la experiencia (**?**). La experiencia se refiere a datos etiquetados (ejemplos) que permiten inferir un modelo estadístico de aprendizaje. Entre los métodos de ML ampliamente utilizados se pueden mencionar las Support Vector Machines (SVMs), árboles de decisión, o bien los modelos gráficos, como las redes neuronales o los métodos generativos, solo por mencionar algunos. Para las tareas de etiquetado en NLP generalmente

se utilizan modelos gráficos supervisados, por ejemplo, modelos ocultos de Markov (*Hidden Markov Models, HMM*).

No obstante, el lenguaje natural es complejo y dinámico, ya que tiene fenómenos que hacen que las tareas de reconocimiento, generación y procesamiento se vuelvan difíciles para las computadoras. Adicionalmente, existen escenarios donde estos métodos no son efectivos como es el caso de las lenguas de bajos recursos, que son lenguas que tienen pocos recursos digitales con los que trabajar. Por ejemplo, si se tienen pocos datos iniciales para el entrenamiento del modelo de aprendizaje las predicciones serán poco precisas o equivocadas. Los bajos recursos son un escenario común en México donde, a pesar de que existe una rica diversidad lingüística, gran parte de las lenguas originarias no poseen contenido web ni publicaciones digitales y por tanto carecen también de tecnologías del lenguaje. El escenario mencionado anteriormente supone un reto para los métodos de aprendizaje convencionales, que requieren de grandes cantidades de datos de entrenamiento para funcionar correctamente. Por lo tanto, es un importante reto de investigación desarrollar aproximaciones que funcionen con lenguas de escasos recursos. En particular, en este trabajo nos enfocamos en el glosado automático del otomí, una lengua con gran riqueza morfológica y con escasez de recursos digitales.

El glosado puede ser un primer paso para el desarrollo de más tecnologías del lenguaje; no solo para el otomí, que presenta un grado de extinción acelerada

Diego: ¿Esta cita donde la puedo encontrar?

Vic: Se refiere a la información de la Comisión nacional para el Desarrollo de pueblos Indígenas, pero ahora es el IMPI https://www.gob.mx/inpi

(CDI), sino para las 68 agrupaciones lingüísticas que se hablan en México.

1.3. Objetivo

Diseñar e implementar un etiquetador morfológico para el otomí basado en técnicas de Procesamiento del Lenguaje Natural (*Natural Language Processing, NLP*) con Aprendizaje de Máquina (*Machine Learning, ML*).

En particular, se hará énfasis en métodos de aprendizaje estructurado débilmente supervisados. Específicamente, se aplicará *Conditional Random Fields (CRF)* para etiquetado morfológico (glosado) del otomí, una lengua de bajos recursos.

1.4. Hipótesis

Se espera obtener un modelo que produzca glosa para el otomí, generada automáticamente, con base en el entrenamiento con pocos ejemplos previamente etiquetados. Al obtener una buena exactitud en la predicción automática de glosa se apoyaría a los anotadores humanos a reducir trabajo repetitivo y exhaustivo. Además, se espera obtener avances de una metodología adaptable a un mayor número de lenguas mexicanas. Sería deseable que esta metodología experimental pueda ser replícale en otras lenguas habladas en México.

2 Avances en etiquetadores automáticos

2.1. Marco teórico

2.1.1. Natural Language Processing (NLP)

2.1.2. Etiquetadores

2.1.3. Machine Learning (ML)

2.1.4. Modelos gráficos

Los límites de los modelos gráficos en para bajos recursos

En este capítulo se explicará qué ventajas tienen los *Conditional Random Fields (CRF)* sobre otros modelos de aprendizaje, se mencionan formalmente los elementos fundamentales que describen los *CRF's*.

En lingüística computacional una tarea de interés es el procesamiento estadístico del lenguaje natural, en particular, el etiquetado y segmentación de secuencias de datos. En ese sentido, es habitual la utilización de **modelos generativos**, cómo los *Hidden Markov Models (HMMs)*, o **modelos condicionales**, como los *Maximum Entropy Markov Models (MEMMs)*.

Por una parte, los modelos generativos intentan modelar una probabilidad conjunta $P(x, y)$ sobre observaciones y etiquetas. Para definir esta probabilidad conjunta se necesita enumerar todas las observaciones posibles. Las limitantes de este enfoque son de diversas índoles como las grandes dimensionalidades en el vector de entrada X, la dificultad de representar múltiples características que interactúan unas con otras y dependencias complejas que

hacen la construcción de la distribución de probabilidad un problema intratable con un enfoque computacional.

Por otro lado, una solución a las limitantes de los modelos generativos es un modelo condicional. Estos modelos no son tan estrictos como los primeros al momento de asumir independencias en las observaciones. Los modelos condicionales especifican la probabilidad de posibles etiquetas dada una secuencia de observación.

Consecuencia de lo anterior, no se gasta esfuerzo en modelar las observaciones, dado que en al momento de realizar pruebas estas observaciones son fijas. Segundo, la probabilidad condicional puede depender de características arbitrarias y no dependientes de la secuencia de observación sin forzar al modelo a tomar en cuenta la distribución de estas características, permitiendo que el modelo sea tratable (Lafferty et al., 2001).

Un ejemplo de estas ventajas con los *MEMMs* que son modelos secuenciales de probabilidad condicional. Sin embargo, estos modelos y otros que son no generativos, de estados finitos y que son clasificadores basados en el estado siguiente comparten una debilidad llamada *label bias problem*. Lafferty et al. (2001) define que existe el *label bias problem* cuando "las transiciones que dejan un estado compiten solo entre sí, en lugar de entre todas las demás transiciones en el modelo".

Dado que las transiciones son las probabilidades condicionales de los siguientes posibles estados una observación puede afectar cuál será el estado siguiente sin tomar en cuenta que tan adecuado será este. Por tanto, se tendrá un sesgo en los estados con menos transiciones de salida.

2.1.5. Conditional Random Fields

Como menciona Sutton et al. (2012) modelar las dependencias entre las entrada puede conducir a modelos intratables, pero ignorar estas dependencias puede reducir el rendimiento.

Dado el que problema abordado en este trabajo, dónde se requiere del etiquetado de secuencias y es en contexto de bajos recursos lingüísticos, se hace necesario utilizar un enfoque más conveniente.

Los *Conditional Random Fields (CRFs)* son un framework para la creación de modelos probabilístico utilizado en técnicas de aprendizaje estruc-

turado. Tienen las ventajas de los *MEMMs* y, en principio, solucionan el *label bias problem*. El framework tiene un solo modelo exponencial para la probabilidad conjunta de todas las secuencias de las etiquetas de salida dada la secuencia de observación. En contraste los *MEMMs* usan modelos exponenciales para cada probabilidad condicional de los estados siguientes dado el estado actual.

Formalmente Lafferty et al. (2001) definen los *CRFs* como a continuacion se enuncia:

Definición 1. Sea $G = (V, E)$ una gráfica tal que $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, entonces esa $\mathbf{Y}$ es indexada por los vertices de G. Entonces $(\mathbf{X}, \mathbf{Y})$ es un conditional random field en caso de que las variables aleatorias $\mathbf{Y}$ se condicionen por $\mathbf{X}$, la variable aleatoria $\mathbf{Y}_v$ cumple la *propiedad de Markov* con respecto a la gráfica: $p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \sim v)$, dónde $w \sim v$ significa que w y v son vecinos en G.

En esta tesis, para el modelado de secuencias, se utiliza la forma más sencilla de la gráfica G dónde es una cadena simple o línea. Esto quiere decir que $G = (V = \{1, 2, ...m\}, E = \{(i, i+1)\})$. A este tipo de *CRFs* se les conoce como *linear-chain CRFs*. Como menciona Lafferty et al. (2001) "si la gráfica $G = (V, E)$ de $\mathbf{Y}$ es un árbol (del cual una cadena es el ejemplo más sencillo), los *cliques* son los límites y vertices. Entonces, por el teorema de los *random fields* (Hammersley y Clifford, 1971), la distribución conjunta sobre las etiquetas de secuencias $\mathbf{Y}$ y $\mathbf{X}$ tiene la forma:

$$\mathrm{p}_\theta(y|x) \propto \exp\left(\sum_{e \in \mathbf{E}, k} \lambda_k f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v \in \mathbf{V}, k} \mu_k g_k(v, \mathbf{y}|_v, \mathbf{x})\right) \tag{2.1}$$

De la ecuación se destacan f_k y g_k que representan las *feature functions*. Estas están definidas y son fijas. Las *feature functions* de está tesis serán descritas más adelante.

2.2. Estado del arte

Los *CRFs* han sido utilizados para la clasificación de regiones en una imagen, estimar el puntaje en un juego de Go, segmentar genes en una hebra de ADN y análisis sintáctico de lenguaje natural en un texto por mencionar algunas (Sutton et al., 2012).

2.2.1. Trabajos sobre bajos recursos

Con base en el trabajo previo de (Moeller y Hulden, 2018) donde se utilizan técnicas de NLP y ML para tratar para el idioma Lezgi se plantea como hipótesis que dado el tamaño del corpus y la glosa que contiene

se obtendrá texto correctamente glosado con una precisión de al menos 80 %.

3 Etiquetador morfológico para el otomí (Metodología)

En este capitulo se describe el corpus utilizado en este trabajo, se mostrará la arquitectura propuesta para la generación automática de glosa para el idioma otomí. Adicionalmente, se explicará el diseño e implementación del *pipeline* que incluye, entre otras cosas, la determinación de las *feature functions*.

Los *CRFs* muestran claras ventajas sobre otros métodos de aprendizaje basados en gráficas. Su habilidad de tomar las virtudes de los modelos generativos y de los modelos condicionales presentan a este *framework* como una opción para el contexto de los bajos recursos digitales que nos impone, como se describirá más adelante, el tamaño del corpus. En ese sentido, los *CRFs* se utilizaron para predecir secuencias de etiquetas, utilizando el enfoque del aprendizaje estructurado, que describen las unidades morfológicas dentro de una palabra, de una variante en particular del otomí.

3.1. Corpus: otomí de Toluca

La clasificación lingüística introduce al otomí dentro de las lenguas otomianas, las cuales a su vez pertenecen a la rama otopame de la familia otomangue (Barrientos López, 2004). Cada variante muestra particularidades fonológicas, morfológicas, sintácticas y léxicas. En el tratamiento de textos por medio de técnicas de *NLP* se requiere que estos textos estén normalizados y homogéneos. Lo anterior propicia la obtención del mejor desempeño posible en los diversos métodos de aprendizaje automático. Mas adelante se describirá el proceso de preprocesamiento aplicado al corpus.

Se utilizó un corpus en otomí que, además, cumple la característica de

Categoría	**Cuenta**
Tokens (POS)	8578
Tipos (POS)	44
Tokens (Glosa)	14477
Tipos (Glosa)	112
Total de oraciones etiquetadas	1786

Tabla 3.1: Tamaño del corpus

Textos	**Número**
Narrativos	32
Dialogados	4
Total de textos	36

Tabla 3.2: Textos del corpus

estar glosado. Se trabajó con la variante del otomí de Toluca de la región de San Andrés Cuexcontitlan.

Esta tesis recogió un corpus basado en el trabajo de Lastra y de Suárez (1992) titulado *El otomí de Toluca* y que a su vez fue etiquetado y glosado manualmente por el lingüista Víctor Germán Mijangos de la Cruz[1]. Este corpus es un subconjunto del corpus paralelo español-otomi que se encuentra en la plataforma web Tsunkua[2]. Además, se agregaron 81 lineas de casos poco usuales y que son fenómenos poco frecuentes y por tanto particularmente difíciles de predecir. El subconjunto del corpus utilizado en la sección experimental está descrito en la tabla 3.1, donde se encuentra el tamaño de las etiquetas POS y el tamaño de la glosa y en la tabla 3.2, donde se puede ver los tipos de textos presentes en el corpus.

Los textos que componen el corpus fueron construidos a partir de las aportaciones de diez hablantes distintos de entre diez y setenta y tres años, de los cuales, siete son de sexo femenino y tres masculino (Lastra y de Suárez, 1992).

[1]TODO: Liga del repo
[2]https://tsunkua.elotl.mx/

v	obl	det	cnj	dem
unkwn	n	neg	p.loc	prt
conj.adv	dim	gen	cond	it
lim	aff	loc	dec	conj
cord	san	cnj.adv	regular/v	adv
adj				

Tabla 3.3: Tipos de etiquetas *POS*

Etiqueta	**Significado**	**Etiqueta**	**Significado**
v	verbo	obl	oblicuo
det	determinante	cnj	conjunción
dem	demostrativo	unkwn	desconocido
n	sustantivo	neg	negativo
p.loc	partícula locativoa	prt	partícula
conj.adv	conjunción adversativa	dim	diminutivo
gen	genitivo	cond	condicional
it	iterativo	lim	limitativo
aff	afirmativo	loc	locativo
dec	decimal	conj	conjunción
cord	coordinación	cnj.adv	conjunción adversativa
regular/v	verbo regular		

Tabla 3.4: Descripción de etiquetas *POS*

Los tipos de etiquetas *POS* presentes en el corpus se pueden observar en la tabla 3.3. Dentro del conjunto de estas las etiquetas encontramos un subconjunto con elementos como buena.vista, nada.más y zapata. Estos ejemplos son una forma de glosa muy común en el uso lingüístico y no están incluidos en la tabla 3.3 pues son descriptivos en sí mismos. Presentamos una descripción de las etiquetas en la tabla 3.4.

Vic: Esta forma de glosar es común en el uso lingüístico, no vale la pena presentarlo en la tabla pues son descriptivos por sí mismos.

Para las etiquetas de Glosa nos basamos en las reglas de **?** desarrolladas

stem	det	3.cpl	psd	lim
prag	3.icp	lig	det.pl	1.icp
3.pot	ctrf	1.pot	pl.exc	1.cpl
dem	1.pss	dim	pl	1.obj
ila	2.icp	1.prf	3.cnt	3.obj
loc	mod	1.cnt	3.pls	prt
it	dual.exc	3.prf	3.icp.irr	3.pss
2.pss	1.enf	med	dual	p.loc
2.cnt	2	3.imp	int	neg
1.icp.irr	1.cpl.irr	2.obj	aum	1.pls
2.cpl	2.prf	gen	com	2.pot
adj	cond	3.cpl.irr	1.sg	encl
3.sg	3.pss.pl	spt	1.irr	2.enf
conj.adv	caus	con	chico	eh
comp	prf	dist mov	3.irr	det.dem
dcl	nom	2.icp.irr		

Tabla 3.5: Glosa

por el departamento de lingüística del Instituto Max Planck y el Departamento de lingüística de la Universidad de Leipzing. El estándar consiste en diez reglas para la sintaxis y la semántica de glosas interlineales. y un apéndice con un lexicón propuesto de etiquetas de categorías abreviadas (**?**). Si bien las reglas cubren parte de las necesidades lingüísticas en el glosado de textos, tambien, son flexibles y se pueden agregar o modificar las convenciones dependiendo de las necesidades.

Los tipos de etiquetas de glosa presentes en el corpus se encuentran en la tabla 3.5. Los números al inicio de algunas etiquetas significan las personas gramaticales. Existen combinaciones de varias etiquetas que son separadas por puntos. Por ejemplo, pl.exc es una combinación de las etiquetas "pluralƶ .^{ex}clusivo". El significado para cada glosa se muestra en la tabla 3.6. En esta tabla se omitieron las variaciones de etiquetas con personas gramaticales para compactarla.

Glosa	Significado	Glosa	Significado
stem	base	ctrf	contrafactual
cpl	completivo	dem	demostrativo
icp	incompletivo	dim	diminutivo
pot	potencial	ila	ilativo
ctn	continuativo	mod	modo
prf	perfecto	loc	locativo
pls	pluscuamperfecto	prt	partícula
irr	irrealis	it	iterativo
imp	imperativo	enf	enfático
psd	pasado	neg	negativo
pl	plural	int	interrogativo
sg	singular	aum	aumentativo
ex	exclusivo	gen	genitivo
pss	posesivo	com	comitativo
obj	objeto	adj	adjetivo
med	voz media	encl	enclítico
dual	número dual	enf	enfático
det	determinante	caus	causativo
lim	limitativo	comp	comparativo
lig	ligadura	dcl	declarativo
prag	partícula pragmática		

Tabla 3.6: Descripción de Glosa

Vic: También pongo la tabla para las glosas, pero la reduzco, los números 1,2 y 3 son las personas. Creo que se pueden quitar las etiquetas que se ya están en la tabla anterior. Tamboén hay que acomodar las tablas para que se vean mejor

Etiqueta	Tokens
v	2596
obl	2447
det	975
cnj	837
dem	543
unkwn	419
n	273
neg	178
p.loc	81
prt	49

Tabla 3.7: POS (Primeros 10)

3.1.1. Tokens

3.1.2. Distribución de etiquetas

3.2. Arquitectura

Para esta tesis proponemos una arquitectura de aprendizaje estructurado débilmente supervisado utilizando el método gráfico *CRF* que permitirá la predicción de secuencias que describen las unidades morfológicas (glosa) dentro de una palabra en otomí. Ya que el resultado esperado es la generación de etiquetas que, en principio, dependen unas de otras, un método basado en gráficas como los *CRFs* fué el adecuado.

Los *CRFs* predicen las secuencias de glosa, que será la salida Y dadas las observaciones X (texto previamente glosado). Puntualmente, se utiliza el modelo **Markov CRF de primer orden con *features* de estado y de transición**. Adicionalmente, es utilizado el algoritmo de aprendizaje de **Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)** como se mencionó en la sección 2.1.

El objetivo de esta arquitectura es realizar un correcto etiquetado automático de glosa para la lengua otomí, en particular la variante del otomí de Toluca, utilizando técnicas de aprendizaje estructurado débilmente su-

Etiqueta	Tokens
stem	7527
det	733
3.cpl	450
psd	418
lim	374
prag	362
3.icp	346
lig	289
det.pl	271
1.icp	270

Tabla 3.8: Glosa (Primeros 10)

pervisado. Se definieron de aprender un conjunto de *feature functions* que describen TODO el contexto y brindan información útil para la fase de entrenamiento.

El *pipeline* de aprendizaje semi-supervisado para la generación de glosa para

el otomí de Toluca se describe en las secciones siguientes.

Ximena

Tener bien claro cuál es nuestra hipótesis, separarnos un poco del trabajo de lezgi

3.2.1. Codificación y preprocesamiento

Se obtiene el corpus de un archivo de texto plano. Cada renglón de este archivo es una oración en otomí con glosa. Además, se tiene una etiqueta *POS* por cada palabra. Las frases están estructuradas en forma de listas que contienen otras listas validas para el lenguaje `Python`.

La glosa esta presente por cada fragmento de las posibles palabras en otomí. Ejemplificando, la frase "hi tótsogí"(*No lo he dejado*)

Vic: No tendría porque ir en negritas, puede ir entre ” o en cursivas. También convendría poner su traducción ’No lo he dejado’

se representa en el corpus como se muestra en el ejemplo 3.2.1. Por último, debido a que este tipo de lenguas contienen más vocales . Cuando se codificaban en cadenas eran separadas por el lenguaje de programación oca-

Diego

TODO: Citar blog de elot. Incluir tabla de vocales

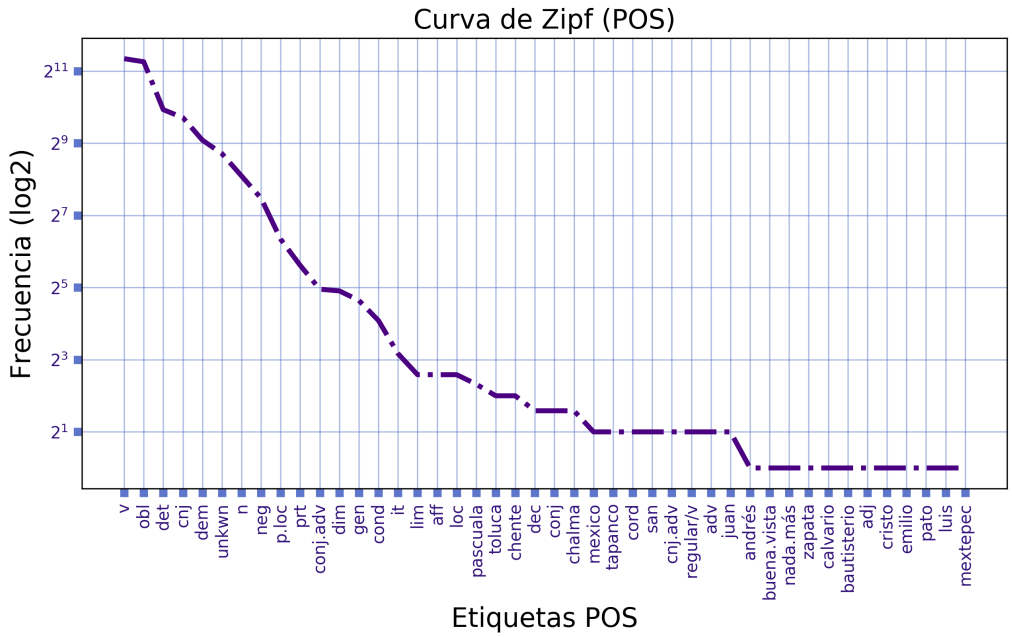

Figura 3.1: Distribución de etiquetas POS

sionando un etiquetando de dichas marcas que por si solas no tiene sentido. Entonces, fue necesario modificar algunas letras compuestas en otomí. Las equivalencias de tales modificaciones pueden observarse en la tabla 3.9

Ejemplo 3.2.1. `[[['hi', 'stem'], 'neg'], [['tó', '1.prf'], ['tsogí', 'stem'], 'v']]`

Vic: En este caso, el ejemplo quedaría mejor más cerca de la referencia

Entonces, la estructura de las listas, por renglón, tiene la estructura `[[[letras, glosa], [letras, glosa], ..., POS],...]`. Una vez que se obtuvo el corpus de entrenamiento se le aplicó preprocesamiento. El preprocesamiento consistió en asociar, a cada letra de cada palabra, dos elementos; la etiqueta *POS* y una *Bio Label* correspondiente a esa letra.

Retomando el ejemplo 3.2.1 y después de aplicar el preprocesamiento el

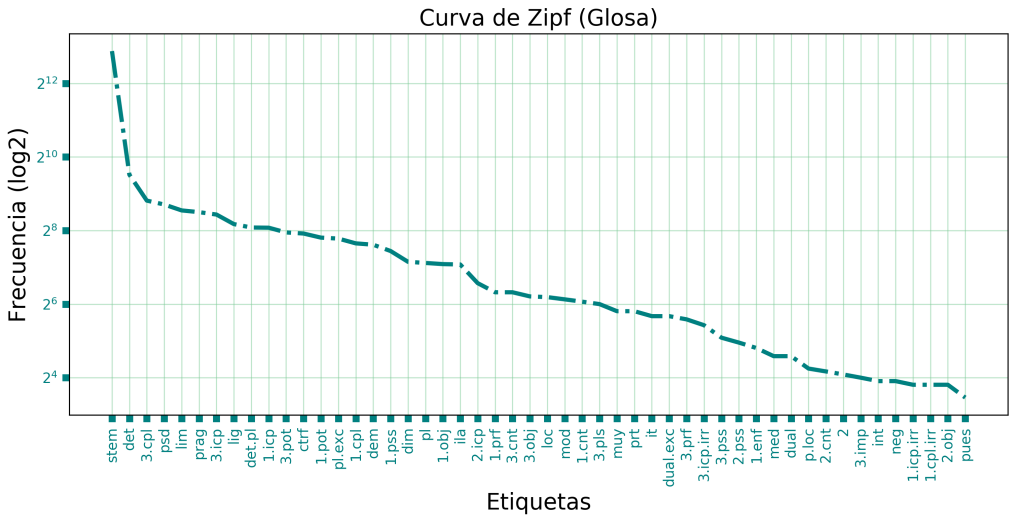

Figura 3.2: Distribución de glosa (primeras 50)

otomí	**equivalencia**
a̲	α
e̲	ϵ
i̲	ι
u̲	μ

Tabla 3.9: Letras en otomí modificadas

resultado sería el que se muestra en el ejemplo 3.2.2. Cabe señalar que las *Bio Labels* asignadas dependieron de la posición de la letra como se explico en la sección 2.1.

Ejemplo 3.2.2. ['ó', 'v', 'I-1.prf'],
['t', 'v', 'B-stem'],
['s', 'v', 'I-stem'],
['o', 'v', 'I-stem'],
['g', 'v', 'I-stem'],
['í', 'v', 'I-stem']]]

Con las palabras etiquetadas a nivel de letra se obtuvo un conjunto de entrenamiento y un conjunto de pruebas. Por una parte, el conjunto de entrenamiento provee la capacidad de observar los ejemplos y así generar un modelo de aprendizaje. Por otro lado, el conjunto de pruebas nos permitió medir que tan la precisión que tuvo el modelos etiquetando frases no vistas. Es importante destacar que estos dos conjuntos estuvieron completamente separados ya que mezclar el conjunto de entrenamiento y de pruebas pudo habernos llevado a una precisión erronea y sesgada.

Previo al entrenamiento se construyeron las *feature functions* con el conjunto de entrenamiento. En ese sentido, por cada letra etiquetada de las palabras en otomí se tuvo una *feature function* y por cada *feature function* se tiene una *Bio Label* asociada. La construcción de estas funciones será descrita a profundidad en la subsección 3.2.3.

Los *CRFs* recibieron como entrada las *feature functions*, representadas por un vector X y sus respectivas *Bio Labels*, representadas por un vector y, asociadas con cada *feature function* con concordancia uno a uno respetando la posición.

3.2.2. Hiperparametros

El entrenamiento consistió en la búsqueda de un modelo que maximiza el logaritmo de verosimilitud con el método de aprendizaje *L-BFGS*. Es necesario entonces definir los parámetros del algoritmo de maximización. Utilizamos *Elastic Net* con los valores de regularización, necesarios para evitar problemas como el *overfitting*, L1 + L2 y el máximo de iteraciones. Ya definidos los hiperparametros se ejecutó la fase de entrenamiento.

Una vez terminada la fase de entrenamiento y construido el modelo de aprendizaje se puso a prueba con el conjunto destinado a este propósito y que fue previamente obtenido. Similar a la fase de entrenamiento las entradas fueron las *feature functions* y sus respectivas *Bio Labels*. Utilizando el modelo de aprendizaje se etiquetó cada letra con base en la información de las *feature functions*. Las etiquetas generadas fueron comparadas con las reales y se obtuvo la exactitud del modelo.

Consideramos exitosa la predicción si se logra maximizar la correcta clasificación de las secuencias de salida. Para determinar si la predicción fue

exitosa se utilizaron técnicas típicas de *ML* como *K-folds* que consiste en tomar K fragmentos de los datos de entrada y utilizarlos para probar el modelo. Además de reportar la precisión se obtuvo el *recall* y *F-score*.

3.2.3. Feature functions

La extracción de estas características es importante porque capturar fenómenos lingüísticos necesarios para que la estructura de la lengua se pueda plasmar en el modelo de aprendizaje. Estas características están capturando, entres otras cosas, el contexto de la palabra y es importante para predecir la morfología. Para construir las *feature functions* se necesita el corpus glosado y etiquetado a nivel de letra. Se extrajeron las siguientes características para cada letra en el corpus:

- bias: Esta característica captura la proporción de una etiqueta dada en el conjunto de entrenamiento. Ayuda a tomar en cuenta que algunas etiquetas son poco usuales y otras muy usuales.
- letterLowercase: Toma la letra y la convierte a minúsculas. Es importante para la creación del modelo tener en cuenta la letra que se esta viendo en un momento determinado para las predicciones posteriores.
- prevpostag: Toma la etiqueta POS previa (Si existe)
- nxtpostag: Toma la etiqueta POS siguiente (Si existe)
  - Es conveniente y muy útil tomar en cuenta las etiquetas *POS* ya que brindan información gramatical de la palabra que se observa en ese momento. Tal información se basa en la morfología y algunas veces en la sintaxis de la lengua.
- BOS: Indica el inicio de la frase
- EOS: Indica el fin de la frase
- BOW: Indica el inicio de la palabra
- EOW: Indica el final de la palabra

  - Indicar el inicio y fin de frase y palabras otorga la capacidad de ver que tipo de palabras se está viendo en ese momento. Por ejemplo, una palabra que esté justo al inicio de una oración en general será un ..., mientras la palabra al final de una oración probablemente será un ...

Diego
Duda de que estructuras son comúnes en otomí

- letterposition: Indica la posición de la letra en la palabra
- prev4letters: Toma las cuatro letras previas (Si existen)
- prev3letters: Toma las tres letras previas (Si existen)
- prev2letters: Toma las dos letras previas (Si existen)
- prevletter: Toma la letra previa (Si existe)
- nxtletter: Toma la siguiente letra (Si existe)
- nxt2letters: Toma las siguientes dos letra (Si existen)
- nxt3letters: Toma las siguientes tres letra (Si existen)
- nxt4letters: Toma las siguientes cuatro letra (Si existen)

  - La recuperación del varias ventanas contexto son convenientes para del otomí ya que se trata de una lengua aglutinante dónde, en general, las palabras son largas. En particular, la longitud promedio de las palabras en el corpus es de 4.89. Esta característica da la pauta para que la observación del contexto en un determinado momento sea relevante para la construcción del modelo.

Las características mencionadas son información relevante para poder construir un modelo más preciso dadas las estructuras de la lengua. Dichas características pueden o no estar presentes dependiendo de la letra que se esté viendo en ese momento. Por ejemplo, si es la primer letra de una palabra la que se observa no estarán presentes las características prevletter, prev2letters o EOW por mencionar algunas. Características como letterLowercase o bias siempre estuvieron presentes.

Hardware utilizado

En la fase de experimentación fue utilizado el paquete python-crfsuite que es un *binding* de la implementación de TODO:CITA para CRFs para lenguaje de programación *Python* en su versión *3.7*. La fase de experimentación corrió en una maquina con un procesador Intel i7-7700HQ @ 3.800GHz con 16 GB de memoria principal. En promedio una corrida de entrenamiento y evaluación *K-folds* con *k=10* tomó 52 minutos.

4 Resultados

4.1. Corpus de evaluación

4.2. Evaluación

4.3. Análisis de resultados

5 Conclusiones

Bibliografía

Guadalupe Barrientos López. 2004. *Otomíes del Estado de México*.

John M Hammersley y Peter Clifford. 1971. Markov fields on finite graphs and lattices. *Unpublished manuscript* 46.

John Lafferty, Andrew McCallum, y Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data .

Yolanda Lastra y Yolanda Lastra de Suárez. 1992. *El otomí de Toluca*. Instituto de Investigaciones Antropológicas, UNAM.

Sarah Moeller y Mans Hulden. 2018. Automatic glossing in a low-resource setting for language documentation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*. pages 84–93.

Anil Kumar Singh y Samar Husain. 2005. Comparison, selection and use of sentence alignment algorithms for new language pairs. In *Proceedings of the ACL Workshop on Building and using Parallel texts*. Association for Computational Linguistics, pages 99–106.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4(4):267–373.