PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

Corrector ortográfico neuronal para errores ortográficos multilingües adversarios para lenguas amazónicas peruanas

Trabajo de investigación para obtener el grado académico de Magíster en Informática con mención en Ciencias de la Computación que presenta:

Gerardo Cardoso Yllanes

Asesor:

Héctor Erasmo Gómez Montoya

Lima, 2022

Corrector ortográfico neuronal para errores ortográficos multilingües adversarios para lenguas amazónicas peruanas

Gerardo Cardoso*, Erasmo Gómez*, Arturo Oncevay*†
* *Ingeniería Informática, Pontificia Universidad Católica del Perú*
† *School of Informatics, University of Edinburgh*
{gerardo.cardoso, hector.gomez}@pucp.edu.pe, a.oncevay@ed.ac.uk

***Resumen*—Para combatir los ataques de ejemplos adversarios, se propuso implementar un modelo de reconocimiento de palabras y entrenarlo con oraciones creadas a través de diferentes técnicas de generación de data aumentada para cuatro lenguas amazónicas peruanas de pocos recursos: Shipibo-Konibo, Asháninka, Yanesha y Yine. Observamos que, para la gran mayoría de experimentos, el modelo propuesto logró corregir oraciones con palabras con errores ortográficos. Los modelos que fueron entrenados mediante oraciones creadas a través de los canales de errores de ambigüedad fonema-grafema y desnormalización; y, el modelo de ensamble, se desempeñaron mejor al momento de evaluarlos con los corpus creados por profesores de las lenguas. Finalmente, se implementó un prototipo del corrector ortográfico neuronal, en donde se encuentran todos los modelos entrenados en la presente investigación.**

I. Introducción

En el Perú, se hablan oficialmente 48 lenguas indígenas, de las cuales cuatro son lenguas andinas y 44 son lenguas amazónicas [1]. Lamentablemente, varias de estas se encuentran en peligro de extinción, principalmente, debido a la reducida transmisión intergeneracional [2]. Gran parte de las lenguas peruanas son consideradas como lenguas de pocos recursos; esto quiere decir que no cuentan con una descripción o documentación. De hecho, solo alrededor del 50% tiene algún documento almacenado en alguna base de datos [1]. Actualmente, existe poco apoyo estatal basado en políticas de revitalización y promoción de las lenguas peruanas [1]. Bajo ese contexto, para facilitar estas labores de revitalización, existe la necesidad de introducir recursos informáticos. Una de estas necesidades es el desarrollo de un corrector ortográfico para lenguas amazónicas peruanas [3]. Esta tarea consiste en la recolección de recursos bibliográficos; luego, crear un corpus de oraciones sin errores ortográficos. Posteriormente, crear un corpus paralelo con oraciones con errores ortográficos; y, finalmente, implementar un modelo de aprendizaje profundo que permita realizar correcciones ortográficas automáticas y eficientes.

Alva y Oncevay [4], investigadores de la Pontificia Universidad Católica del Perú, realizaron una investigación en la que se exploraron diferentes técnicas de generación de datos artificiales para entrenar un corrector ortográfico basado en un modelo neuronal secuencia-a-secuencia para cuatro lenguas amazónicas peruanas de pocos recursos: Shipibo-Konibo, Ashaninka, Yanesha y Yine . Esta investigación tuvo resultados alentadores sobre la posibilidad de crear correctores ortográficos para lenguas de pocos recursos.

El modelo actual, el cuál será la línea base de la presente investigación, permite la corrección ortográfica automática de texto para cuatro lenguas amazónicas peruanas; sin embargo, existen algunos puntos que no se lograron cubrir, los cuales son los siguientes: la carencia de corpus se traduce en un obstáculo durante el proceso de la elaboración del corrector ortográfico [5]; y no existe un corpus de oraciones sin errores ortográficos y con errores ortográficos creado exclusivamente por profesores de las lenguas para poder evaluar el desempeño del corrector ortográfico neuronal, con ejemplos que se asemejan a los que se encuentran en producción. Es importante mencionar que la recolección de estos corpus para lenguas de pocos recursos, como las lenguas amazónicas del Perú, es complicado y costoso debido a que existen pocos documentos digitalizados en las bases de datos; y para la construcción de estos se necesita de la intervención de lingüistas y profesores de las lenguas [6].

Por otro lado, se ha comprobado que la robustez de un modelo neuronal contra ataques de ejemplos adversarios, los que contienen información de entrada con perturbaciones y tienen como objetivo lograr que los modelos se equivoquen y clasifiquen incorrectamente, se puede mejorar entrenando con una mayor cantidad de datos generados artificialmente [7]. El actual modelo explora diferentes técnicas de generación de datos artificiales basados en proximidad de teclado, ambigüedad de fonema a grafema y similitud de sílabas; sin embargo, es necesario probar otras técnicas de generación de datos artificiales con el objetivo de construir un modelo neuronal más robusto.

Además, si bien es cierto que las redes neuronales profundas han contribuido de manera satisfactoria al desarrollo de diferentes aplicaciones de aprendizaje profundo, los investigadores han demostrado que estas redes neuronales son vulnerables a ataques de ejemplos adversarios [8].

El último punto a considerar es que no existe un corrector ortográfico disponible como servicio para las cuatro lenguas amazónicas peruanas, lo que representa un paso importante para contribuir a la revitalización de estas.

El objetivo de la actual investigación es implementar un

corrector ortográfico neuronal robusto contra ataques de ejemplos adversarios para cuatro lenguas amazónicas peruanas. Para alcanzar dicho objetivo, se propone realizar las siguientes tareas:

En primer lugar, crear un corpus de oraciones sin errores ortográficos y con errores ortográficos, elaborado exclusivamente por profesores de las cuatro lenguas amazónicas peruanas.

En segundo lugar, diseñar algoritmos para generar ejemplos de ataques adversarios a través de diversos canales de errores.

En tercer lugar, diseñar un modelo neuronal robusto contra ataques de ejemplos adversarios. Gracias al continuo desarrollo del área de procesamiento de lenguaje natural [9], se han desarrollado nuevas estrategias para construir una robusta defensa dentro del diseño de la red neuronal con el objetivo de poder crear un corrector ortográfico más eficiente [10].

Por último, implementar un prototipo del corrector ortográfico que se encuentre disponible para los hablantes de las cuatro lenguas amazónicas peruanas. Esta herramienta permitirá la corrección ortográfica automática de texto para las lenguas amazónicas peruanas: Shipibo-Konibo, Asháninka, Yanesha y Yine.

El esquema del documento es el siguiente. En la Sección 2, se revisan las lenguas y conjuntos de datos de la investigación. Luego, en la Sección 3, prevemos el estado del arte, el cual incluye varios artículos científicos relacionados con el problema. Después, se presentan los canales de errores para construir los corpus paralelos en la Sección 4. En la Sección 5, se presentan la experimentación y comparación de resultados. Luego, en la Sección 6, se presenta el prototipo del corrector ortográfico neuronal. Finalmente, se presentan las conclusiones y posibles trabajos futuros en la Sección 7.

II. Lenguas y conjuntos de datos

Similar a la línea base, la investigación se enfoca en la elaboración de un corrector ortográfico neuronal para cuatro lenguas amazónicas peruanas de pocos recursos: Shipibo-Konibo, Asháninka, Yanesha y Yine. Para cada lengua, se heredó un corpus monolingüe de oraciones, extraídos de libros educacionales en sus versiones PDF, de dónde se filtró y seleccionó solo las oraciones que tenían 50 o menos caracteres de longitud.

La tabla I muestra la información del corpus "base" monolingüe de oraciones para las cuatro lenguas amazónicas peruanas.

TABLA I
CORPUS "BASE" MONOLINGÜE DE ORACIONES PARA LAS CUATRO LENGUAS AMAZÓNICAS PERUANAS.

Lengua	Número oraciones	Tamaño vocabulario
Shipibo-Konibo	22,032	22,904
Asháninka	12,629	23,721
Yanesha	13,241	23,626
Yine	7,658	14,142

Posteriormente, se obtiene un corpus monolingüe de oraciones sin errores ortográficos y un corpus paralelo monolingüe tipificado de oraciones con errores ortográficos para las cuatro lenguas amazónicas peruanas; estos fueron creados exclusivamente por profesores de las lenguas. Para su creación, a cada profesor se le envió una lista de palabras únicas y, por cada palabra, crearon una oración sin errores ortográficos que contenga esa palabra o alguna similar, y otra oración con errores ortográficos con base en la oración anterior. De igual manera, por cada palabra con error ortográfico, se etiquetó el tipo de error.

La tabla II muestra la información de los corpus "general" monolingüe de oraciones sin errores ortográficos y con errores ortográficos creado por los profesores, y en la tabla III, el conteo de tipo de errores.

TABLA II
CORPUS "GENERAL" MONOLINGÜE DE ORACIONES SIN ERRORES ORTOGRÁFICOS Y CON ERRORES ORTOGRÁFICOS CREADO POR LOS PROFESORES.

Lengua	Número oraciones (cada uno)	Tamaño vocabulario	
		sin errores	con errores
Shipibo-Konibo	2,936	10,710	13,336
Asháninka	3,544	11,385	13,209
Yanesha	3,490	10,146	12,793
Yine	2,078	6,131	6,710

TABLA III
CONTEO DE TIPO DE ERRORES DEL CORPUS "GENERAL" PARALELO MONOLINGÜE TIPIFICADO DE ORACIONES CON ERRORES ORTOGRÁFICOS.

	Shipibo-Konibo	Asháninka	Yanesha	Yine
Fónico	2,132	1,354	5,540	1,347
Género	142	282	-	1
Tiempo	96	66	-	-
Número	51	111	9	2
Puntuación	47	43	327	-
Acentuación	39	-	238	-
Sintáctico	3,622	1,272	330	3,916
Semántico	517	93	-	-

Luego, se requiere validar si el corrector ortográfico neuronal es capaz de corregir oraciones escritas con una gramática no normalizada a una normalizada de la lengua. Una gramática no normalizada se refiere a la gramática que aún no se ha establecido el alfabeto ni las reglas para su escritura [2]. Para las validaciones se tiene un corpus monolingüe de oraciones escritas con la gramática no normalizada y un corpus paralelo monolingüe de oraciones escritas con la gramática normalizada para las cuatro lenguas amazónicas peruanas. Estos fueron creados exclusivamente por profesores de dichas lenguas. Para su creación, a cada profesor se le envió una lista de palabras únicas y, por cada palabra, crearon una oración escrita con la gramática no normalizada de la lengua que contenga esa palabra o alguna similar, y otra oración escrita con la gramática normalizada con base en la anterior oración. Estos corpus se usan exclusivamente para las pruebas del corrector ortográfico neuronal.

La tabla IV muestra la información de los corpus "normalización" monolingüe de oraciones con la gramática normalizada y no normalizada creado por los profesores.

TABLA IV
CORPUS "NORMALIZACIÓN" MONOLINGÜE DE ORACIONES CON LA GRAMÁTICA NORMALIZADA Y NO NORMALIZADA CREADO POR LOS PROFESORES.

Lengua	Número oraciones (cada uno)	Tamaño vocabulario	
		norm.	no norm.
Shipibo-Konibo	916	3,931	4,279
Asháninka	796	3,124	3,291
Yanesha	754	1,781	1825
Yine	702	1,667	1,710

Finalmente, se tiene un conjunto de corpus que se elaboraron con parte de los resultados de la investigación de Oncevay [11] sobre un traductor de textos de español a lenguas del Perú. Para esta investigación, solo se consideran los resultados de las lenguas Shipibo-Konibo y Asháninka. El primer corpus monolingüe es un conjunto de oraciones que la red neuronal tradujo deficientemente de español a una lengua, y otro es un corpus paralelo monolingüe de oraciones con las referencias correctamente. Estos corpus se usan exclusivamente para las pruebas del corrector ortográfico neuronal, con el objetivo de validar si este es capaz de poseditar, que significa, en el contexto de traducción automática, a la corrección de las oraciones con inferencias deficientes, luego de pasar por el proceso automático de traducción [12].

La tabla V muestra la información de los corpus "posedición" monolingüe con las referencias correctas y traducciones deficientes.

TABLA V
CORPUS "POSEDICIÓN" MONOLINGÜE DE ORACIONES CON REFERENCIAS CORRECTAS Y TRADUCCIONES DEFICIENTES.

Lengua	Número oraciones (cada uno)	Tamaño vocabulario	
		referencias	deficientes
Shipibo-Konibo	500	1,735	1,267
Asháninka	502	1,815	1420

III. ESTADO DEL ARTE

Existen varios artículos que investigan sobre cómo combatir ataques adversarios en el área de procesamiento de lenguaje natural. Jayanthi et al. [13] crearon NeuSpell, una herramienta que recopila diferentes correctores ortográficos en inglés y, también, implementa su propio corrector ortográfico neuronal. En su investigación, resaltan que varios de los correctores ortográficos neuronales no toman en consideración el contexto alrededor de la palabra con error ortográfico, donde proponen entrenar una red neuronal usando errores ortográficos en contexto, que fueron construidos sintéticamente, y representaciones contextuales. Entrenando el modelo con sus ejemplos sintéticos, obtuvieron una mejora del 9% respecto a sus modelos que fueron entrenados con perturbaciones aleatorias a nivel caracter, y una mejora del 3% haciendo uso del contexto. Luego, Jones et al. [10] introducen codificaciones robustas (RobEn), las cuales representan un entorno de trabajo que otorga robustez sin comprometer la arquitectura del modelo. La clave de esto es la función de codificación, la que convierte las oraciones a un espacio de codificación más pequeño. De esta manera, demostraron que los sistemas que utilizan estos encodings confieren robustez mediante un entrenamiento estándar. Por otro lado, Belinkov y Bisk [14] estudiaron cómo los modelos de traducción a nivel de caracteres eran afectados por ruido natural y sintético. Ellos tomaron dos enfoques para aumentar la robustez del modelo: representación invariante de la estructura y entrenamiento adversarial como defensas ante el ruido. En cambio, Danish et al. [15], para combatir errores adversarios, proponen anteponer un modelo de reconocimiento de palabras a un clasificador. Este modelo es una red neuronal recurrente semicaracter, la cual introduce estrategias para manejar palabras raras o no vistas. Con la introducción de este modelo de reconocimiento de palabras ante un clasificador de sentimiento, se pudo restaurar su precisión hasta un 30%.

Con respecto a investigaciones de correctores ortográficos para lenguas amazónicas peruanas, se cuenta con el corrector propuesto por Alva y Oncevay [14] para Shipibo-Konibo. Su desarrollo cuenta con dos pasos: un método automático de silabeo basado en reglas y un grafo. Para finalizar, se encuentra otra investigación por parte de Alva y Oncevay [4], la cuál será la línea base de investigación, en donde implementaron una red neuronal entrenada a través de datos aumentados, los cuales fueron generados a través de diferentes canales de errores: proximidad de teclado, ambigüedad fonema-grafema y similitud de sílaba. Sin embargo, se puede observar que esta red neuronal no cuenta con mecanismos para combatir, de manera efectiva, ejemplos adversarios.

IV. CANALES DE ERRORES

A continuación, se presentan los canales de errores para construir los corpus paralelos de oraciones con errores ortográficos adversarios:

A. Error aleatorio (ErrAle)

En este canal de error, se perturban las palabras a través de alguna de las 4 ediciones a nivel de caracter: insertar (agregar un nuevo caracter), eliminar (remover un caracter), intercambiar (intercambiar de posición a dos caracteres adyacentes de una palabra) o reemplazar (sustituir un caracter por otro de una palabra). Solo se realizan estas perturbaciones a las palabras que tienen una longitud igual o mayor a 3 caracteres. Este canal de error es independiente a la lengua; esto quiere decir que se aplica a las cuatro lenguas amazónicas peruanas.

B. Proximidad de teclado (ProxTec)

Este canal de error se basa en la intuición de una persona que, al momento de tipear una palabra por medio del teclado, puede involuntariamente presionar una tecla cercana a la tecla que quería presionar. Para este algoritmo, se utiliza una estructura de datos tipo diccionario, en donde la llave es una letra y el valor es una lista de letras próximas según sus distribuciones en el teclado. Este canal de error es independiente a la lengua; esto quiere decir que se aplica a las cuatro lenguas amazónicas peruanas. En el algoritmo 1, se encuentra la serie de pasos para generar una palabra con un error ortográfico a través del canal

de error de proximidad de teclado. El algoritmo tiene, como entrada, una palabra sin errores ortográficos y tiene, como salida, una palabra con un error ortográfico.

Algoritmo 1: Proximidad de teclado

Entrada: p
$p =$*"jaweratorin" en Shipibo-Konibo*
Salida: pError
1: $pos = PosiciónAleatoria(p)$
$pos = 6$
2: $car = p[pos]$
$car =$'t'
3: $carProx = CaracterPróximoAleatorio(car)$
$carProx =$'r'
4: $pError = p[0 : pos - 1] + carProx + p[pos + 1 :]$
$pError =$"jaweraroriN"
5: **return** $pError$

C. Ambigüedad fonema-grafema (AmbFonGraf)

Este canal de error se basa en la idea de que existen grafemas que tienen similar pronunciación y, por ende, el oyente puede confundirlas al momento del deletreo. Con la ayuda de los lingüistas de las lenguas, se crea una estructura de datos tipo diccionario, en donde la llave es un grafema y el valor es una lista de posibles confusiones del grafema. Se cuenta con el mapeo para tres lenguas amazónicas peruanas: Shipibo-Konibo, Asháninka y Yanesha. En el algoritmo 2, se encuentra la serie de pasos para generar una palabra con un error ortográfico a través del canal de error de ambigüedad fonema-grafema. El algoritmo tiene, como entrada, una palabra sin errores ortográficos, y tiene, como salida, una palabra con un error ortográfico.

Algoritmo 2: Ambigüedad fonema-grafema

Entrada: p
$p =$*"sewayanon" en Yanesha*
Salida: pError
1: $graf = GrafemaAleatorio(p)$
$graf =$"w"
2: $grafConf = GrafemaConfusiónAleatorio(graf)$
$grafConf =$"hu"
3: $pError = ReemplazarGrafema(p, graf, grafConf)$
$pError =$"sehuayanon"
4: **return** $pError$

D. Similitud de sílaba (SimSíl)

Este canal de error se basa en la idea de que, dentro de una lengua, existen sílabas similares y, por ello, una persona puede confundir una sílaba con otra al momento del deletreo. Alva y Oncevay realizaron un método de silabificación para Shipibo-Konibo [16]; luego de identificar las sílabas, se crea una estructura de datos tipo diccionario, en donde la llave es una sílaba y el valor es una lista de sílabas similares. Con el soporte de los lingüistas, se cuenta con el mapeo para las cuatro lenguas amazónicas peruanas. En el algoritmo 3, se encuentra la serie de pasos para generar una palabra con un error ortográfico a través del canal de error de similitud de sílabas. El algoritmo tiene, como entrada, una palabra sin errores ortográficos, y tiene, como salida, una palabra con un error ortográfico.

Algoritmo 3: Similitud de sílaba

Entrada: p
$p =$*"tenonnalu" en Yine*
Salida: pError
1: $sil = SílabaAleatoria(p)$
$sil =$"no"
2: $silSim = SílabaSimilarAleatoria(sil)$
$silSim =$"co"
3: $pError = ReemplazarSílaba(p, sil, silSim)$
$pError =$"teconnalu"
4: **return** $pError$

E. Desnormalización (Desnorm)

Este canal de error se basa en la idea de que una persona puede hacer uso de una grafía no normalizada en lugar de la normalizada al momento del deletreo. Con la ayuda de los lingüistas de las lenguas, se crea una estructura de datos tipo diccionario, en donde la llave es una grafía de la gramática normalizada y el valor es la grafía equivalente de la gramática no normalizada. Con el soporte de los lingüistas, se cuenta con el mapeo para las cuatro lenguas amazónicas peruanas. En el algoritmo 4, se encuentra la serie de pasos para generar una palabra con un error ortográfico a través del canal de error de desnormalización. El algoritmo tiene, como entrada, una palabra construida con la gramática normalizada, y tiene, como salida, una palabra construida con la gramática no normalizada.

Algoritmo 4: Desnormalización

Entrada: pNorm
$pNorm =$*"piñero" en Asháninka*
Salida: pNoNorm
1: $graf = GrafíaAleatoria(pNorm)$
$graf =$"ñe"
2: $grafNoNorm = GrafíaNoNormalizada(graf)$
$grafNoNorm =$"nie"
3: $pNoNorm =$
$ReemplazarGrafía(p, graf, grafNoNorm)$
$pError =$"piniero"
4: **return** $pNoNorm$

V. Experimentación y resultados

Para los experimentos, utilizamos un modelo de reconocimiento de palabras implementado por Pruthi et al. [15] para combatir errores ortográficos adversarios. Este es una red

neuronal recurrente semicaracter (ScRNN) basada en un modelo diseñado por Sakaguchi et al. [17], el cual fue inspirado en la idea psicolingüística de que la lectura humana es resiliente a la transposición de caracteres internos. La red neuronal procesa una oración con palabras con errores ortográficos y predice una oración con palabras sin errores ortográficos. Cada palabra que ingresa a la ScRNN se representa por un vector, en donde este a su vez está compuesto por tres partes, la primera y tercera consisten en la representación One-Hot del primer y último caracter de la palabra, y la segunda está compuesto por la representación de una bolsa de los caracteres internos de la palabra. Cada vector alimentó a la célula BiLSTM de la red neuronal en cada paso; la salida de esta fue tomada como entrada para la capa softmax, la que finalmente tuvo como salida una palabra del vocabulario. La configuración de los hiperparámetros del modelo ScRNN está detallada en el apéndice.

Como empleamos un modelo de red neuronal secuencia-a-secuencia, usamos chrF como métrica de evaluación [18]. También, incluimos el valor ΔchrF, que es la diferencia entre los valores chrF después y antes de evaluarlos en los modelos.

A continuación, detallamos los experimentos que se realizaron para la actual investigación:

A. Línea base

Experimento A través de los corpus "base" monolingües de oraciones para las cuatro lenguas amazónicas peruanas, se crearon corpus paralelos monolingües de oraciones con errores ortográficos por cada canal de error. Cada palabra con error ortográfico de una oración se generó al aplicar un solo tipo de canal de error. La distribución de los conjuntos de datos para la experimentación es de 93% para entrenamiento, 3.5% para validación, y 3.5% para pruebas. Ninguno de los conjuntos de datos compartió oraciones entre sí. Adicionalmente, para cada lengua, se entrenó un modelo "AmbFonGraf + Desnorm", en el que sus conjuntos de datos estaban compuesto por la suma de los conjuntos de datos de los modelos "AmbFonGraf" y "Desnorm". Luego, se entrenó un modelo "Todos", en el que sus conjuntos de datos estaban compuestos por la suma de los conjuntos de datos de todos los modelos. Finalmente, se implementó un modelo ensamble por votación, en donde se combinó las predicciones de todos los modelos.

Resultados Calculamos los valores chrF y los valores ΔchrF. Los resultados se presentan en la tabla VI. Se pudo observar que los valores ΔchrF son positivos. Esto quiere decir que los modelos lograron corregir palabras con errores ortográficos. Los modelos entrenados a través de los corpus generados por medio de los canales de errores ambigüedad fonema-grafema y desnormalización fueron los que obtuvieron mejores resultados; esto debido a que, para la creación de oraciones con errores ortográficos, utilizan una lista limitada, creando así errores similares en las palabras, por lo que la red neuronal pudo aprender y converger correctamente. De igual forma, para los otros modelos entrenados a través de los corpus generados por medio de los canales de errores de ruido aleatorio, proximidad de teclado y similitud de sílaba, se obtuvieron buenos resultados. Sin embargo, estos no fueron tan buenos como los anteriores debido a que los canales producían palabras con errores distintos y como consecuencia el modelo no pudo converger apropiadamente.

B. Generalización

Experimento Para este experimento, se quiso identificar qué modelos entrenados mediante qué canales de errores se desempeñan mejor al evaluarlos con los corpus "general" creados por los profesores de las cuatro lenguas amazónicas peruanas. Para validación y pruebas, se utilizaron 500 oraciones del corpus "general" cada uno, y para entrenamiento, se utilizó el 100% del corpus "base" más las oraciones restantes del corpus "general". Ninguno de los conjuntos de datos compartió oraciones entre sí. Adicionalmente, para cada lengua, se entrenó un modelo "AmbFonGraf + Desnorm", en el que sus conjuntos de datos estaban compuesto por la suma de los conjuntos de datos de los modelos "AmbFonGraf" y "Desnorm". Luego, se entrenó un modelo "Todos", en el que sus conjuntos de datos estaban compuestos por la suma de los conjuntos de datos de todos los modelos. Finalmente, se implementó un modelo ensamble por votación, en donde se combinó las predicciones de todos los modelos.

Resultados Calculamos los valores chrF y los valores ΔchrF. Los resultados se presentan en la tabla VII. Se pudo observar que los valores ΔchrF son positivos en casi todos los modelos. Esto quiere decir que los modelos pudieron corregir palabras con errores ortográficos. Nuevamente, los modelos entrenados a través de los corpus generados por medio de los canales de errores ambigüedad fonema-grafema y desnormalización obtuvieron buenos resultados. Esto nos muestra que las palabras con errores ortográficos creados por estos canales de errores son similares a los errores ortográficos de los corpus de oraciones creados por los profesores. Adicionalmente, el modelo ensamble obtuvo buenos resultados como consecuencia de la combinación de las predicciones de todos los modelos.

C. Normalización

Experimento Para este experimento, se quiso identificar qué modelos entrenados del experimento "Generalización" se desempeñaban mejor al evaluarlos con los corpus "normalización" creados por los profesores de las cuatro lenguas amazónicas peruanas.

Resultados Calculamos los valores ΔchrF. Los resultados se presentan en la tabla VIII. Se pudo observar que los valores ΔchrF son positivos en casi todos los modelos y como se pretendía obtener, los modelos entrenados a través de los corpus generados por medio de canal de error de desnormalización fueron los que obtuvieron mejores resultados. Este resultado evidencia que las palabras con errores ortográficos creados por estos canales de errores son similares a los errores ortográficos de los corpus "normalización" de oraciones creados por los profesores.

TABLA VI
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "LÍNEA BASE".

chrF	Shipibo-Konibo			Asháninka			Yanesha			Yine		
	train	val	test	train	val	test	train	val	test	train	val	test
ErrAle	88.0	77.5	76.8 (22.8)	92.1	84.2	83.3 (15.9)	89.9	80.4	80.1 (18.0)	90.4	82.3	**81.4 (18.4)**
ProxTec	92.2	81.9	82.4 (33.1)	91.4	82.5	82.8 (18.1)	89.1	81.4	81.7 (23.3)	89.7	81.0	80.3 (20.7)
AmbFonGraf	92.6	89.3	**89.8 (38.0)**	94.7	91.2	**91.1 (14.2)**	90.5	86.7	86.7 (23.4)	-	-	-
SimSíl	87.2	73.3	73.4 (19.3)	91.9	81.0	81.9 (12.8)	90.9	81.1	80.4 (11.4)	90.1	80.4	78.4 (13.9)
Desnorm	93.5	90.4	**90.8 (27.8)**	93.4	90.9	**90.7 (14.8)**	94.6	92.1	**92.4 (15.8)**	92.3	90.1	**88.7 (14.9)**
AmbFonGraf + Desnorm	93.1	89.3	89.7 (32.3)	93.7	90.3	90.3 (13.8)	92.5	88.9	**89.1 (19.6)**	-	-	-
Todos	86.5	78.0	77.9 (23.5)	91.3	83.8	84.0 (13.2)	90.0	82.3	81.8 (16.1)	90.4	82.1	81.2 (16.0)
Ensamble	-	-	71.9 (17.5)	-	-	83.9 (13.1)	-	-	79.8 (14.2)	-	-	79.3 (14.2)

TABLA VII
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "GENERALIZACIÓN".

chrF	Shipibo-Konibo			Asháninka			Yanesha			Yine		
	train	val	test	train	val	test	train	val	test	train	val	test
ErrAle	89.6	85.1	85.3 (5.4)	89.3	89.3	88.5 (0.2)	88.9	74.4	75.2 (4.0)	90.2	85.9	85.6 (6.6)
ProxTec	92.0	85.2	85.8 (5.9)	89.4	89.6	89.2 (0.8)	88.7	75.3	76.5 (5.4)	88.6	85.7	85.2 (6.2)
AmbFonGraf	92.5	88.3	88.4 (8.5)	93.9	90.0	89.1 (0.8)	90.3	76.0	**77.0 (5.9)**	-	-	-
SimSíl	81.7	83.5	84.1 (4.2)	88.1	88.7	87.8 (-0.5)	89.2	75.1	75.2 (4.0)	89.6	85.6	84.8 (5.9)
Desnorm	93.3	88.0	**88.5 (8.6)**	93.1	90.4	**89.6 (1.3)**	94.5	75.9	76.9 (5.7)	92.2	86.6	**86.4 (7.5)**
AmbFonGraf + Desnorm	92.4	88.1	88.1 (8.2)	93.7	89.8	89.0 (0.6)	91.6	75.8	76.8 (5.7)	-	-	-
Todos	84.7	84.5	84.7 (4.9)	86.5	87.9	86.6 (-1.7)	86.7	74.0	74.7 (3.6)	90.6	84.1	83.9 (4.9)
Ensamble	-	-	**88.7 (8.8)**	-	-	**89.8 (1.4)**	-	-	**77.4 (6.3)**	-	-	**86.2 (7.3)**

TABLA VIII
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "NORMALIZACIÓN".

chrF	Shipibo-Konibo	Asháninka	Yanesha	Yine
ErrAle	88.9 (2.1)	75.7 (1.9)	64.6 (3.4)	72.6 (1.0)
ProxTec	88.4 (1.6)	74.0 (0.1)	64.8 (2.7)	73.4 (1.8)
AmbFonGraf	91.3 (4.5)	78.9 (5.1)	71.0 (8.9)	-
SimSíl	87.9 (1.2)	75.4 (1.6)	63.6 (1.4)	70.9 (-0.7)
Desnorm	**92.4 (5.6)**	**80.4 (6.6)**	**72.3 (10.2)**	**80.4 (8.7)**
AmbFonGraf + Desnorm	91.3 (4.5)	**78.9 (5.1)**	**73.2 (11.1)**	-
Todos	88.6 (1.9)	76.8 (3.0)	68.2 (6.1)	75.7 (4.0)
Ensamble	**91.7 (5.0)**	78.0 (4.2)	67.9 (5.8)	**76.6 (5.0)**

D. Tipos de errores

Experimento Para este experimento, se quiso identificar qué tipos de errores ortográficos se desempeñan mejor al evaluarlos en los dos mejores modelos del experimento "generalización". Para cada lengua y cada tipo de error, se creó un corpus de oraciones que contiene una sola palabra con un error ortográfico de un tipo.

Resultados Calculamos los valores ΔchrF. Los resultados se presentan en la tabla IX. Se pudo observar que los valores ΔchrF son positivos, independientemente del tipo de error ortográfico.

E. Palabras nuevas

Experimento Para este experimento, se quiso medir el desempeño de los modelos del experimento "generalización" al evaluarlos con oraciones sin errores ortográficos que contengan palabras nuevas o no vistas durante el entrenamiento. Para cada lengua, se creó un corpus de 50 oraciones del conjunto de pruebas de oraciones sin errores ortográficos que contengan palabras nuevas o no vistas durante el entrenamiento.

Resultados Calculamos los valores ΔchrF. Los resultados se presentan en la tabla X. Se pudo observar que los valores ΔchrF son ligeramente negativos, esto quiere decir que los modelos descorrigen en un porcentaje muy pequeño las oraciones sin errores ortográficos.

F. Posedición

Experimento Para este experimento, se quiso identificar qué modelos entrenados del experimento "Generalización" se desempeñaban mejor al evaluarlos con los corpus "posedición".

Resultados Calculamos los valores de ΔchrF. Los resultados se presentan en la tabla XI. Los valores ΔchrF son negativos, esto quiere decir que no pudieron corregir efectivamente las oraciones con traducciones deficientes.

VI. PROTOTIPO DEL CORRECTOR ORTOGRÁFICO

Como mencionamos anteriormente, no existe un corrector ortográfico disponible como servicio para las lenguas amazónicas peruanas. La elaboración de esta herramienta representaría un paso importante para la revitalización de

TABLA IX
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "TIPO DE ERRORES".

chrF	Shipibo-Konibo		Asháninka		Yanesha		Yine	
	Desnorm	Ensamble	Desnorm	Ensamble	Desnorm	Desnorm	Desnorm	Ensamble
Fónico	97.3 (2.5)	**97.6 (2.8)**	**97.4 (0.7)**	97.1 (0.5)	95.8 (1.7)	96.1 (1.9)	94.8 (1.8)	94.8 (1.8)
Género	**97.7 (3.5)**	**97.7 (3.6)**	95.8 (1.3)	95.2 (0.6)	-	-	100.0 (2.1)	100.0 (2.1)
Tiempo	**97.5 (3.4)**	97.5 (3.5)	97.0 (1.6)	**96.3 (1.0)**	-	-	-	-
Número	97.2 (3.3)	97.0 (3.0)	96.9 (1.2)	96.2 (0.4)	**93.7 (3.6)**	92.4 (2.3)	**100.0 (7.6)**	**100.0 (7.6)**
Puntuación	96.6 (2.8)	97.0 (3.2)	96.9 (0.7)	95.9 (-0.3)	**89.9 (3.5)**	90.4 (3.9)	-	-
Acentuación	96.7 (2.6)	96.9 (2.8)	-	-	89.7 (3.2)	**90.9 (4.3)**	-	-
Sintáctico	97.0 (2.3)	97.2 (2.5)	**97.8 (0.7)**	97.6 (0.5)	90.4 (3.3)	**91.3 (4.3)**	**96.3 (3.0)**	**96.3 (3.0)**
Semántico	96.7 (2.5)	96.8 (2.6)	97.1 (1.2)	**96.5 (0.6)**	-	-	-	-

TABLA X
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "PALABRAS NUEVAS".

chrF	Shipibo-Konibo	Asháninka	Yanesha	Yine
ErrAle	96.3 (-3.7)	96.5 (-3.5)	96.1 (-3.9)	97.4 (-2.6)
ProxTec	94.7 (-5.3)	96.7 (-3.3)	95.9 (-4.1)	98.1 (-1.9)
AmbFonGraf	97.4 (-2.6)	96.7 (-3.3)	96.2 (-3.8)	-
SimSíl	94.7 (-5.3)	95.3 (-4.7)	96.8 (-3.2)	96.9 (-3.1)
Desnorm	**98.2 (-1.8)**	**97.1 (-2.9)**	**97.3 (-2.7)**	**98.7 (-1.3)**
AmbFonGraf + Desnorm	97.5 (-2.5)	96.6 (-3.4)	96.3 (-3.7)	-
Todos	96.1 (-3.9)	95.4 (-4.6)	96.4 (-3.6)	97.8 (-2.2)
Ensamble	**98.9 (-1.1)**	**97.7 (-2.3)**	**98.3 (-1.7)**	**99.1 (-0.9)**

TABLA XI
RESULTADOS DE LOS VALORES CHRF (Y ΔCHRF) DEL EXPERIMENTO "POSEDICIÓN".

chrF	Shipibo-Konibo	Asháninka
ErrAle	44.2 (-2.7)	37.4 (-2.5)
ProxTec	44.6 (-2.3)	38.3 (-1.6)
AmbFonGraf	45.3 (-1.5)	37.9 (-2.0)
SimSíl	44.9 (-1.9)	37.3 (-2.6)
Desnorm	45.7 (-1.2)	**38.9 (-1.0)**
AmbFonGraf + Desnorm	**45.7 (-1.1)**	37.8 (-2.1)
Todos	45.3 (-1.6)	37.6 (-2.3)
Ensamble	**45.9 (-0.9)**	**38.7 (-1.2)**

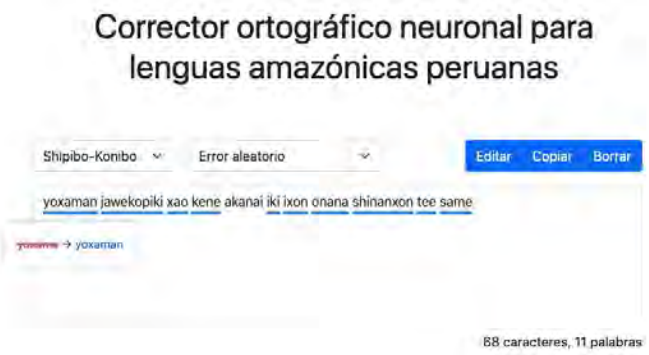

Fig. 1. Prototipo del corrector ortográfico neuronal para las cuatro lenguas amazónicas peruanas.

las mismas. Por lo tanto, se creó un prototipo del corrector ortográfico neuronal para las cuatro lenguas amazónicas peruanas: Shipibo-Konibo, Asháninka, Yanesha y Yine.

El prototipo es una web desarrollada mediante el framework Flask [19], donde se cargaron todos los modelos neuronales que se entrenaron en las diferentes experimentaciones de la actual investigación.

En la interfaz del prototipo, el usuario puede elegir la lengua y el modelo que la aplicación utilizará para realizar las correcciones. Luego, el usuario introduce una oración con errores ortográficos. Después, la web envía los parámetros y la oración a corregir a la aplicación. Finalmente, este retorna una oración sin errores ortográficos en la que se puede observar todas las correcciones que el corrector neuronal realizó.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

En la mayoría de experimentos, el modelo neuronal propuesto logró corregir oraciones con palabras con errores ortográficos. Por un lado, todos los modelos de las cuatro lenguas amazónicas peruanas, entrenados y evaluados exclusivamente mediante los corpus de la línea base de investigación, pudieron mejorar los valores de la métrica chrF. Por otro lado, la mayoría de los modelos que fueron evaluados por oraciones creadas exclusivamente por los profesores de las lenguas pudieron mejorar los valores de la métrica chrF. Los modelos que fueron entrenados mediante oraciones creadas a través de los canales de errores de ambigüedad fonema-grafema y desnormalización, presentaron buenos resultados en las experimentaciones. Adicionalmente, el modelo ensamble por votación también obtuvo buenos resultados en las diversas experimentaciones. Luego, se pudo observar que los modelos pueden corregir las palabras con errores ortográficos, independientemente del tipo de error ortográfico. Después, los modelos descorrigen en un porcentaje muy pequeño oraciones sin errores ortográficos. Con lo que respecta al desempeño del modelo neuronal, al evaluarlo con el corpus de "posedición",

el corrector ortográfico no pudo corregir o poseditar eficientemente las oraciones con traducciones deficientes. Esto se puede atribuir a que el vocabulario y caracteres que utiliza es diferente al de los corpus de oraciones con los que fueron entrenados los modelos. Para mejorar el rendimiento, se propone entrenar el modelo mediante oraciones que provengan de la misma fuente. Finalmente, para cada lengua y canal de error, se creó un modelo independiente. Lo que se plantea como trabajo futuro es crear un único modelo multilingüe y multicanal que pueda corregir oraciones con errores ortográficos de diversas lenguas y de diferentes canales de errores, lo que se traduciría a que se requiera menos esfuerzo en el entrenamiento, despliegue y mantenimiento en general en comparación a tener varios modelos especializados para cada combinación de lengua y canal de error.

AGRADECIMIENTOS

Los autores están agradecidos con Candy Angulo, Jaime Montoya, Gema Silva y Adriano Ingunza, lingüistas de las cuatro lenguas amazónicas peruanas, por su participación en la etapa inicial del proyecto, proporcionando información relacionada con las lenguas. También, realizamos un agradecimiento especial a los profesores Jovita Vásquez, Saúl Escobar, Delio Siticonatzi, Esaú Zumaeta, Didier López, Juan López, Nimia Acho y Remigio Zapata y al especialista Rubén Ruiz, integrantes de la Universidad Católica Sedes Sapientiae, por su participación en la elaboración de los corpus de oraciones para el presente proyecto.

REFERENCIAS

[1] R. Zariquiey, H. Hammarström, M. Arakaki, A. Oncevay, J. Miller, A. García, and A. Ingunza, "Obsolescencia lingüística, descripción gramatical y documentación de lenguas en el Perú: hacia un estado de la cuestión," *Lexis*, vol. 43, no. 2, pp. 271–337, 2019.

[2] Ministerio de Educación, *Documento Nacional de Lenguas originarias del Perú*. primera edición ed., 2013.

[3] Grupo de Reconocimiento de Patrones e Inteligencia Artificial Aplicada Pontificia Universidad Católica del Perú, Lima, Perú, A.-P. Galarreta, A. Melgar, and A. Oncevay-Marcos, "Corpus Creation and Initial SMT Experiments between Spanish and Shipibo-konibo," in *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*, pp. 238–244, Incoma Ltd. Shoumen, Bulgaria, Noviembre 2017.

[4] C. Alva and A. Oncevay, "Data augmentation and subword segmentation for spell-checking in amazonian languages," (Lima, Perú), Julio 2021.

[5] A. Imankulova, T. Sato, and M. Komachi, "Improving Low-Resource Neural Machine Translation with Filtered Pseudo-Parallel Corpus," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, (Taipei, Taiwan), pp. 70–78, Asian Federation of Natural Language Processing, Noviembre 2017.

[6] H. E. G. Montoya, K. D. R. Rojas, and A. Oncevay, "A Continuous Improvement Framework of Machine Translation for Shipibo-Konibo," in *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, (Dublin, Ireland), pp. 17–23, European Association for Machine Translation, Agosto 2019.

[7] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial Example Generation with Syntactically Controlled Paraphrase Networks," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1875–1885, Association for Computational Linguistics, Junio 2018.

[8] Z. Meng and R. Wattenhofer, "A Geometry-Inspired Attack for Generating Natural Language Adversarial Examples," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 6679–6689, International Committee on Computational Linguistics, 2020.

[9] G. Bustamante, A. Oncevay, and R. Zariquiey, "No Data to Crawl? Monolingual Corpus Creation from PDF Files of Truly low-Resource Languages in Peru," in *Proceedings of the 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 2914–2923, European Language Resources Association, Mayo 2020.

[10] E. Jones, R. Jia, A. Raghunathan, and P. Liang, "Robust Encodings: A Framework for Combating Adversarial Typos," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 2752–2765, Association for Computational Linguistics, 2020.

[11] A. Oncevay, "Peru is Multilingual, Its Machine Translation Should Be Too?," in *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, (Online), pp. 194–201, Association for Computational Linguistics, 2021.

[12] H. Moon, C. Park, J. Seo, S. Eo, and H. Lim, "An Automatic Post Editing With Efficient and Simple Data Generation Method," *IEEE Access*, vol. 10, pp. 21032–21040, 2022.

[13] S. M. Jayanthi, D. Pruthi, and G. Neubig, "NeuSpell: A Neural Spelling Correction Toolkit," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 158–164, Association for Computational Linguistics, 2020.

[14] Y. Belinkov and Y. Bisk, "Synthetic and Natural Noise Both Break Neural Machine Translation," Febrero 2018.

[15] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating Adversarial Misspellings with Robust Word Recognition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5582–5591, Association for Computational Linguistics, 2019.

[16] C. Alva and A. Oncevay, "Spell-Checking based on Syllabification and Character-level Graphs for a Peruvian Agglutinative Language," in *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, (Copenhagen, Denmark), pp. 109–116, Association for Computational Linguistics, 2017.

[17] K. Sakaguchi, K. Duh, M. Post, and B. Van Durme, "Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network," *arXiv:1608.02214 [cs]*, Febrero 2017. arXiv: 1608.02214.

[18] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, (Lisbon, Portugal), pp. 392–395, Association for Computational Linguistics, 2015.

[19] "Welcome to Flask — Flask Documentation (2.0.x)."

APÉNDICE

Configuración del modelo Para la implementación del modelo, utilizamos una red neuronal recurrente semicaracter propuesto en el trabajo de Pruthi et al. [15]. Los hiperparámetros para actual investigación son los siguientes:

- Arquitectura: LSTM bidireccional
- Número de capas ocultas: 50
- Tamaño del vocabulario: 5,000 para Shipibo-Konibo, Asháninka y Yanesha; y, 3,000 para Yine
- Número de épocas: 100
- Tamaño del batch: 32
- Optimizador: Adam
- Tasa de aprendizaje: 0.001
- Pérdida: Entropía cruzada categórica
- Métrica: chrF