

# Traductor lenguas indígenas colombianas

Laura Rodriguez<sup>1</sup>, Alma Trigos<sup>1</sup>, Miguel Ruiz<sup>1</sup>, Carlos Escobar<sup>1</sup>

<sup>1</sup>Departamento de Ingeniería de Sistemas y Computación, Universidad de los Andes, Bogotá, Colombia.

Contributing authors: [la.rodriguez@uniandes.edu.co](mailto:la.rodriguez@uniandes.edu.co);  
[a.trigos@uniandes.edu.co](mailto:a.trigos@uniandes.edu.co); [m.ruizt@uniandes.edu.co](mailto:m.ruizt@uniandes.edu.co);  
[cd.escobarv1@uniandes.edu.co](mailto:cd.escobarv1@uniandes.edu.co);

## Resumen

El presente trabajo presenta distintos modelos para resolver la tarea de traducción de dos lenguas indígenas colombianas con el fin de observar cual modelo es el mejor y qué factores pueden impactar en los resultados. En donde se puede evidenciar y concluir que un factor importante en tareas de traducción es el tamaño del dataset usado.

**Keywords:** Traductor, Lenguas de bajo recurso, Transformers, Wayuunaiki, Nasa Yuwe

## 1. Introducción

En la actualidad vivimos en una sociedad con una gran diversidad de idiomas y culturas. Muchos de estos provienen de grupos y sociedades con una cantidad reducida de personas, lo cual genera un gran riesgo a que la historia de estos se pierda, siendo un impacto importante para la historia de la humanidad haciendo que a futuro no se pueda estudiar dichas sociedades y culturas, que en su mayoría son grupos indígenas, según la UNESCO [1]. Es de allí la importancia de tener modelos y sistemas que permitan al ser humano a traducir las lenguas de bajos recursos, en este caso, lenguas indígenas a lenguas con mayor número de hablantes que tienen poca probabilidad de perderse o de desuso, como lo es el español.

Con el fin de ayudar a preservar este tipo de lenguas, actualmente se están desarrollando distintas investigaciones que buscan crear traductores eficientes. Una de estas investigaciones que se pueden encontrar es *Enriching Wayúunaiki-Spanish Neural Machine Translation with Linguistic Information* [2], la cual trata acerca del primer

sistema con modelos de traducción automática neuronal (NMT, por sus siglas en inglés) para el lenguaje Wayúunaiki y explora algunas formas de mejorar la traducción de este con el fin de dar bases y fomentar dicha investigación. Sin embargo, no se logran buenos resultados de traducción, pero se logra evidenciar el sesgo de contexto que se llega a tener debido a que mucho del corpus actual se obtiene de textos religiosos.

Otra investigación importante en este campo se encuentra entre el repositorio de la universidad, la cual tiene como título *Machine Translation Strategies for Low-Resource Colombian Indigenous Languages* [3] en la que se implementa una arquitectura Transformer y se usan distintas combinaciones de estrategias para encontrar de qué forma se puede obtener mejores resultados en la tarea de traducción de lenguas indígenas.

Siguiendo con la idea anterior, en el presente documento se prueban 4 distintas arquitecturas preentrenadas que puedan ser útiles para realizar traducción de una lengua indígena colombiana al español con el fin de analizar qué factores pueden impactar en la eficiencia de estos en dicha tarea.

## 2. Metodología

Como metodología de investigación, se realizaron dos pasos importantes. En primer lugar, se recolectó información en español y Wayuunaiki que se procesó para posteriormente realizar distintas arquitecturas con el fin de observar la diferencia de desempeños y los distintos factores que puedan afectar este. A continuación, se va a extender los pasos realizados en la presente investigación.

### 2.1. Datos: Recolección y procesamiento de la información

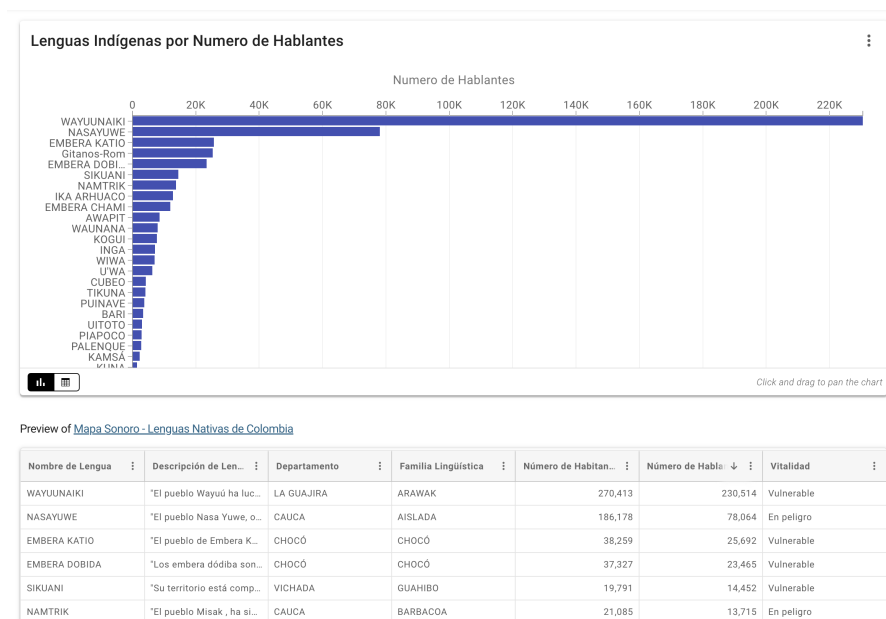
En Colombia existe una gran variedad de lenguas indígenas. A continuación se presenta una tabla que las incluye:

Tipo de Lengua	Ejemplos
Lenguas Indígenas	Achagua, Andoque, Awapit, Bará, Barasano, Barí Ara, Bora, Cabiari, Carapana, Carijona, Cocama, Cofán, Cuiba, Curripaco, Damana, Desano, Embera, Ette Naka, Hitnu, Guayabero, Ika, Inga, Kakua, Kamsá, Kichwa, Kogui, Koreguaje, Kubeo, Kuna, Tule, Macuna, Miraña, Muinane, Namtrik, Nasa-Yuwe, Nonuya, Nukak, Ocaina, Piapoco, Piaroa, Piratapuyo, Pisamira, Puinave, Sáliba, Sikuani, Siona, Siriano, Taiwano, Tanimuca, Tariano, Tatuyo, Tikuna, Tinigua, Tucano, Tucuná, Tuyuca, Uitoto, Uwa, Wanano, Wayuunaiki, Wounaan, Yagua, Yanuro, Yuhup, Yukpa, Yuruti
Lenguas Criollas	Palenquero de San Basilio, Creole de San Andrés y Providencia
Otras Lenguas	Romaní, Lengua de señas colombiana
Variaciones Regionales	Costeño, Paisa, Pastuso, Rolo, etc.

**Cuadro 1** Tipos de Lengua y Ejemplos

Basados en las estadísticas encontradas en el conjunto de datos públicos de datos.gov.co [4], se encuentra que las lenguas con mayor número de hablantes son

**Wayuunaiki y Nasa Yuwe** (ver Figura 1. Dada esta premisa, se asume que son los lenguajes en los que la disponibilidad de contenido digital podría ser mayoritaria, haciendo plausible la obtención de textos procesados o que se puedan procesar manual o automáticamente.



**Figura 1** Visualización Mapa Sonoro

### 2.1.1. Conjunto de Datos en Wayuunaiki

Wayuunaiki (código ISO 639-3 guc) es la lengua de los indígenas Wayuu. Este grupo étnico es originario del departamento de La Guajira, en el norte de Colombia, y también tiene presencia en el país vecino, Venezuela.

Ahora bien, para la selección de este lenguaje, y siguiendo las estadísticas de número de hablantes, se realiza la búsqueda para conjuntos de datos paralelos, textos digitalizados y registros. En esto, se emplean métodos de búsqueda como Google, Bing, Google Scholar, bibliotecas nacionales, incluyendo registros en la biblioteca UniAndes, como también búsquedas apalancadas de agentes inteligentes. Con esto se logran rescatar conjuntos de datos y textos en español y wayuunaiki basados en la biblia, la constitución y diccionarios. Con base en estas fuentes encontradas, se construye un conjunto de datos híbrido que incluye conjuntos de concursos de NLP [5], conjuntos libres en bases de datos de investigación [6] [7], y diccionarios en la web procesados mediante web-scraping [8].

Con base en estos conjuntos de datos recuperados, se utiliza un script de python para procesar y consolidarlos en un dataset que incluye 1.508.456 (66.982 únicas) palabras en español y un total de 119.808 registros de pares de palabras y sentencias como se muestra en la figura 2.

guc_spa_dataset		
✓ 0.0s		
	spa	guc
0	ve	o'unaa
1	vete	pu'unaa
2	vaya	pu'unaa ma
3	vayase	pu'unaa'ma yaaje
4	hola	jamaan
...	...	...
119803	Por lo tanto, evaluemos con detenimiento el ti...	Müsüjese'e waneeküinjatüin tü washaitakat, tü...
119804	Él usa su organización para darnos consejos cu...	Jamüshija'a nia, nüküjün wamüin kojutüinjanai...
119805	CÓMO APOYAR A QUIENES SIRVEN EN OTROS PAÍSES	WAKAALIINJA NA A'YATAAKANA SULU'U WANEE MMA NA...
119806	¿Qué deben hacer los padres si quieren tener b...	¿Kasa naa'inrüinjatka na kachonshiikana ne'ree...
119807	¿Qué ayudó al pueblo de Jehová a mantener viva...	¿Kasa akaaliinjaka na judiökana chayaa Babilonia?

119808 rows × 2 columns

**Figura 2** Muestra Conjunto de Datos Wayuunaiki

### 2.1.2. Conjunto de Datos en Nasa Yuwe

Nasa Yuwe (código ISO 639-3 pbb) es la lengua de los Páez, pueblo originario del departamento de Cauca, en el suroeste de Colombia.

Ahora bien, siguiendo la misma estrategia de búsqueda para el conjunto de wayuunaiki, se encuentran recursos más escasos y con dificultades de procesamiento, por lo que el resultado es un conjunto mucho menor, basado en la constitución de Colombia para algunos artículos y palabras importantes [9], procesado mediante técnicas de ORC con python y la librería tesseract, más algo de ajustes y arreglos manuales, como también diccionarios [10] y conjuntos libres en bases públicas [11]. Después de procesados y organizados, se encuentra un conjunto de datos con 10.858 (4075 únicas) palabras en español y 3794 registros. En la figura 3 se aprecia una muestra del conjunto de datos resultante

spa_pbb_dataset		
✓ 0.0s		
	spa	pbb
0	Artículo 1. Colombia es un Estado social de de...	F'i'n'i pe'la 1. Kulubiyate c'hab wala kiwe' 1'...
1	Artículo 2. Son fines esenciales del Estado: s...	F'i'n'i pe'la 2. Naa c'hab wala kiwete npicthé...
2	Artículo 7. El Estado reconoce y protege la di...	F'i'n'i pe'la 7. Naa Kulubiyate Ec Ne'hwe's'a' ...
3	Artículo 8. Es obligación del Estado y de las ...	F'i'n'i pe'la 8. Naa Kulubiyate npicthé'we's'ma...
4	Artículo 10. El castellano es el idioma oficia...	F'i'n'i pe'la 10. Waas yuwe' Kulubiyate nasa h'...
...	...	...
3789	muchedumbre	nasa kuhsa
3790	territorio propio de poder, mando indígena	nasasa (ikah) kiwe
3791	parte de un todo que contiene diversos element...	f'id'
3792	recurrir en alguna de las forma de protección	ya'nwewen'i
3793	persona que lleva la palabra, mensaje	Yuwe Dukhsa

3794 rows × 2 columns

**Figura 3** Muestra Conjunto de Datos Nasa Yuwe

## 2.2. Arquitecturas

En Transfer Learning, se utiliza un modelo de traducción preentrenado como punto de partida, luego, sin pasos adicionales, se cambia el idioma objetivo a otro idioma y se ajusta finamente el modelo. Se espera que el preentrenamiento mejore el rendimiento en los idiomas de destino. La implementación de Transfer Learning utilizada para los modelos consistirá en reemplazar algunos de los pesos preinicializados aleatorios para cada capa por los pesos ya entrenados de los modelos preentrenados.

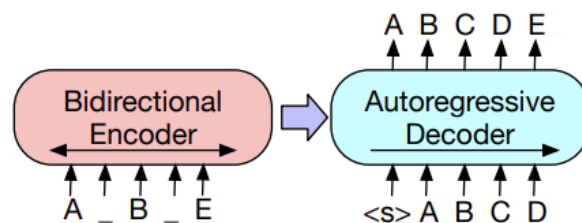
Se escogieron 5 arquitecturas diferentes de modelos ya preentrenados para comparar su eficiencia en la tarea de traducción y se van a explicar cada una a continuación.

### 2.2.1. BART

Modelo BART preentrenado en el idioma inglés. Fue introducido en el artículo "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension" por [12].

Esta es una arquitectura seq2seq estándar, es decir, es un encoder-decoder el cual contiene un encoder bidireccional como lo es BERT y como decoder contiene uno autorregresivo como GPT, la única diferencia es que en la parte del decoder se modifica las funciones de activación ReLU a GeLUs[12]. BART se pre entrena mediante (1) la corrupción de texto con una función de enmascaramiento arbitraria y (2) el aprendizaje de un modelo para reconstruir el texto original.

Adicionalmente, como es especialmente eficaz cuando se ajusta finamente para la generación de texto (por ejemplo, resumen, traducción), y funciona bien para tareas de comprensión (por ejemplo, clasificación de texto, respuesta a preguntas), fue elegido para ser un modelo preentrenado que funcione como base para este proyecto.



**Figura 4** Arquitectura de Bart [12]

### 2.2.2. Llama2

El modelo de LLaMA fue introducido en el artículo "LLaMA: Open and Efficient Foundation Language Models" por [13]. Para la primera versión de LLaMA, se entrenaron cuatro tamaños de modelo: 7, 13, 33 y 65 mil millones de parámetros. No obstante, para efectos de este trabajo, el modelo preentrenado utilizado fue el de 7 millones de parámetros.

Adicionalmente, la arquitectura utilizada por LLaMA utiliza es la transformer. No obstante, existen pequeñas diferencias arquitectónicas en comparación con GPT-3. LLaMA:[13]

- Utiliza la función de activación SwiGLU en lugar de ReLU.
- Emplea embeddings posicionales rotativos en lugar de embeddings posicionales absolutos.
- Utiliza la normalización de capa de raíz cuadrada de la media cuadrática en lugar de la normalización de capa estándar.

### 2.2.3. T5

El modelo de T5 fue introducido en el artículo: "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" por [14]. T5 es un modelo codificador-decodificador preentrenado en una mezcla de tareas no supervisadas y supervisadas mediante un formato texto a texto, donde cada tarea se convierte a dicho formato. El pre entrenamiento incluye tanto entrenamiento supervisado como auto entrenamiento. El entrenamiento supervisado se realiza en tareas específicas proporcionadas por los benchmarks GLUE y SuperGLUE (convirtiéndolas en tareas texto a texto como se explicó anteriormente).

El auto entrenamiento utiliza tokens corruptos, eliminando aleatoriamente el 15 % de los tokens y reemplazándolos con tokens individuales centinelas (si varios tokens consecutivos se marcan para eliminación, todo el grupo se reemplaza con un solo token centinela). El input del codificador es la frase corrupta, el input del decodificador es la frase original y el objetivo son los tokens eliminados delimitados por sus tokens centinelas.

T5 utiliza embeddings escalares relativos. El relleno de la entrada del codificador puede realizarse tanto a la izquierda como a la derecha. Este modelo fue utilizado en 2 tamaños: el base y el largo, para poder evidenciar las diferencias si se cambiaba su tamaño. Estos modelos solo se diferencian en la magnitud y cantidad de datasets con la que fue entrenado

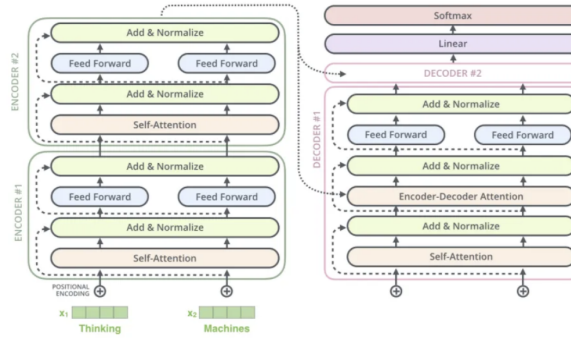


Figura 5 Arquitectura de T5 [15]

#### 2.2.4. MT5

El modelo fue presentado originalmente en el artículo: "mT5: A massively multilingual pre-trained text-to-text transformer" por [16].

El mT5 es una variante multilingüe del modelo T5 que fue preentrenado en un conjunto de datos que abarca más de 101 idiomas y contiene entre 300 millones y 13 mil millones de parámetros. La arquitectura y el entrenamiento del modelo seguido para el mT5 emulan de cerca los del T5. Este modelo de inteligencia artificial aporta diversos tipos de información entre lenguajes similares, lo que beneficia a los idiomas con pocos recursos y permite el procesamiento de lenguajes sin necesidad de entrenamiento específico. La idea de utilizarlo es para ver como mejora sus resultados, siendo que es un modelo multi lenguaje.

### 2.3. Métricas

Posiblemente, la métrica más representativa e importante para los sistemas de traducción es BLEU, por lo que esta fue la utilizada en este trabajo.

#### 2.3.1. BLEU

Esta métrica es una de las más usadas en lo que se refiere a la traducción de máquina, básicamente, utiliza la frecuencia de n-gramas en la frase traducida propuesta por la máquina y la compara con la frase o frases objetivo de referencia que se tengan. La fórmula es la siguiente: [17]

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

Acá,  $w_n$  representa los pesos y  $BP$  es una posible penalización si la respuesta es más larga que el vocabulario de la traducción de referencia


### 2.4. Entrenamiento

El proceso de entrenamiento se desarrolló de forma iterativa, utilizando las herramientas de entrenamiento y fine-tuning de las librerías de Hugging Face. Se crearon dos conjuntos de datos a alto nivel ajustados a la estructura estándar requerida por las arquitecturas mencionadas, i.e., procesamiento para traducción y para instrucción. Estos datasets se publican en Hugging Face y se utilizan como la fuente de datos para él fine-tuning.

Para los modelos transformers de traducción se utilizó una estructura de diccionario anidado con claves como el nombre del lenguaje y el contenido con el texto en cada lenguaje.

Para el ajuste fino de Llama 2, se ajustó el conjunto de manera que realice la traducción por medio de instrucción usando las etiquetas [INST] [/INST]

Se desarrollaron dos scripts que iterativamente se fueron mejorando y estandarizando, hasta lograr una definición base en la cual solo hace falta cambiar los parámetros de lenguaje objetivo y modelo a usar. La definición de hiperparámetros se

id int64	translation dict
	
15,577	{ "guc": "yüleeshaatasü ma'in", "spa": "esta muy fragil" }
39,641	{ "guc": "müsü sünüiki tü wiwüliakat süchiki tia chi wayuu nnojotkai kasain anain apüla tü nünoulakat nu'upala maleiwa watüjaa aa'ulu sünain manoujain nia süka nnojölün naa'inrüin tü...
100,781	{ "guc": "Jamüsüja'a süchikijee tü Segunda Guerra Mundial münakat, ayatüsia ouktün wainma wayuu sutuma tü atkawaakat.", "spa": "7 El fin de la Segunda Guerra Mundial no trajo paz absoluta." }
102,002	{ "guc": "Ai, mototooirüje'e taa'in, Eleena tanülia.", "spa": "Está sonando su teléfono." }
20,339	{ "guc": "nnojoishi pia wa'leewanüin", "spa": "no soy vuestro amigo" }
99,774	{ "guc": "Waa'inrüle tia, aneena waya.", "spa": "Si todos respetamos esos límites, todos podemos disfrutar de verdadera libertad." }

**text**  
string · lengths



<s>[INST] Traduce de Español a Wayuu: En la iglesia Dios ha puesto, en primer lugar, apóstoles; en segundo lugar, profetas; en tercer lugar, maestros; luego los que hacen milagros; después los que tienen dones para sanar enfermos, los que ayudan a otros, los que...

<s>[INST] Traduce de Español a Wayuu: Obedeciendo a una revelación, fui y me reuní en privado con los que eran reconocidos como dirigentes, y les expliqué el evangelio que predico entre los gentiles , para que todo mi esfuerzo no fuera en vano. [/INST] Otta...

<s>[INST] Traduce de Español a Wayuu: »¡Ay de vosotros, maestros de la ley y fariseos, hipócritas! Dais la décima parte de vuestras especias: la menta, el anís y el comino. Pero habéis descuidado los asuntos más importantes de la ley, tales como la justicia, la...

<s>[INST] Traduce de Español a Wayuu: Por eso, dejando a un lado las enseñanzas elementales acerca de Cristo , avancemos hacia la madurez. No volvamos a poner los fundamentos, tales como el arrepentimiento de las obras que conducen a la muerte, la fe en Dios,...

<s>[INST] Traduce de Español a Wayuu: Cuando Israel estaba a punto de morir, mandó llamar a su hijo José y le dijo: -Si de veras me quieres, pon tu mano debajo de mi muslo y prométeme amor y lealtad. ¡Por favor, no me entierres en Egipto! Cuando vaya a descansar...

deja estática en consideración de tasa de aprendizaje o numero de épocas para poder tener resultados comparables sobre la misma linea base.

Los modelos se entrenan usando el servidor de GPU compartido de la universidad y los procesos de entrenamiento se registran en Weights & Biases y el reporte se puede ver en este enlace [SPA-GUC/PBB Fine Tuning](#).

Tanto los datasets creados como los modelos entrenados se pueden revisar y consumir desde [Hugging Face](#) y el codigo para reproducir los resultados se encuentra en el [repositorio del proyecto](#).

### 3. Resultados y discusión

A continuación, se presentan los resultados para cada uno de los modelos probados para ambos lenguajes, posteriormente, se hace una comparación y evaluación del desempeño general de los modelos y una comparación entre ellos y los lenguajes y finalmente se comparan los resultados del presente trabajo con resultados de trabajos relacionados mencionados en la sección 1.



### 3.1. Resultados Wayuunaiki

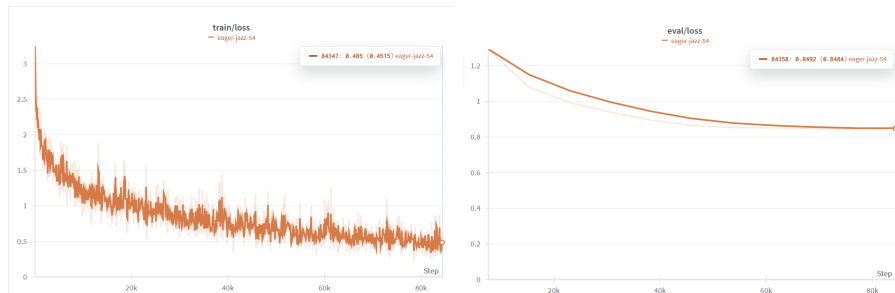
#### 3.1.1. Bart

El modelo estuvo entrenándose por 11h 50m 2s, inicialmente se propuso para que fuera entrenado por 15 épocas, sin embargo, el modelo paró el entrenamiento producto del early stopping en la época número 11, por lo que se puede mostrar que Bart es una arquitectura más sencilla y no pudo mejorar mucho más. En la tabla 2, se puede observar los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos observar que el valor máximo al que se puede llegar es de 3,22, sacrificando un poco del training loss.

Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1	1.555600	1.293041	1.236800	18.944400
2	1.460400	1.077988	1.976100	18.449300
3	0.807600	0.992761	2.238700	18.297000
4	1.191300	0.939769	2.608400	18.208700
5	0.853200	0.894718	2.580900	18.246900
6	0.623400	0.864938	2.737600	18.284200
7	0.798900	0.853499	2.841500	18.228300
8	0.628700	0.851173	2.906100	18.174400
9	0.742900	0.847097	2.976700	18.411500
10	0.358500	0.844153	3.155100	18.266500
11	1.019500	0.848429	3.228900	18.277100

**Cuadro 2** Resultados modelo Bart para Wayuunaiki

Adicionalmente, se puede observar en las imágenes 6, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 0,4 para los datos de entrenamiento y del 0,8, para los datos de validación.



**Figura 6** Resultados función loss Bart Wayuunaiki. a) Datos de entrenamiento, b) Datos de validación

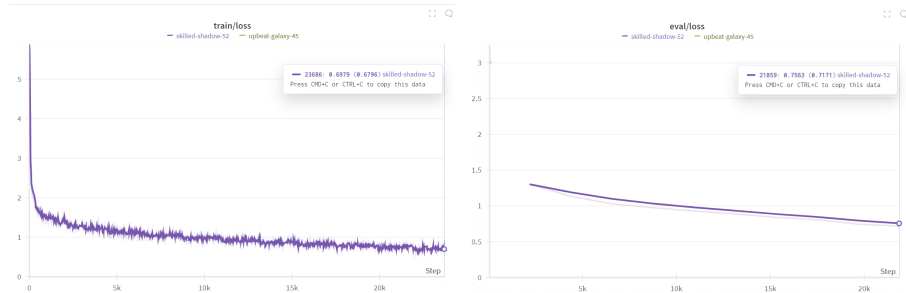
### 3.1.2. Llama2

El modelo estuvo entrenándose por 1d 23h 15m 46s, el modelo paró el entrenamiento producto del early stopping en la época número 10. En la tabla 11, se puede observar los resultados por cada una de las épocas del training y validation loss. Para la métrica de BLEU se calculó el promedio para este modelo, obteniendo 0,13479.

Epoch	Training Loss	Validation Loss	Bleu
1	1.384	1.302397	-
2	1.1099	1.13581	-
3	1.0063	1.030212	-
4	0.9946	0.975233	-
5	0.874	0.929642	-
6	0.8133	0.888141	-
7	0.8264	0.840037	-
8	0.7105	0.806645	-
9	0.7245	0.745025	-
10	0.7235	0.717127	-

**Cuadro 3** Resultados modelo Llama2 para Wayuunaiki

En las gráficas que se pueden observar en las imágenes 7, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 0,66 para los datos de entrenamiento y del 0,71, para los datos de validación, mostrando así un buen rendimiento para el modelo pues no se nota ninguna clase de overfitting presente.



**Figura 7** Resultados función loss Llama Wayuunaiki. a) Datos de entrenamiento, b) Datos de validación

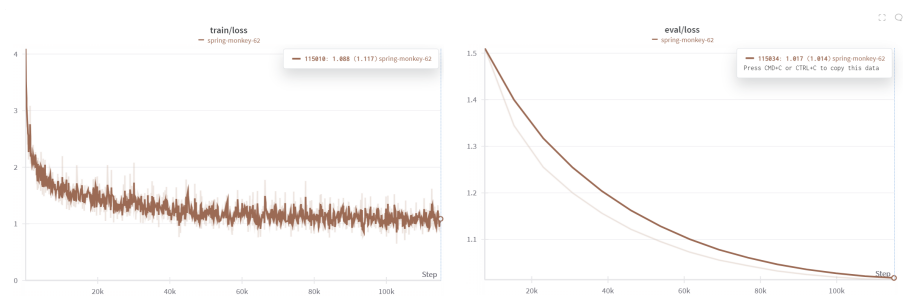
### 3.1.3. T5 base

Este modelo fue entrenado por 1d 13h 26m, y se puede observar que no paró debido al earlystopping, sino que completó las 15 épocas de entrenamiento inicialmente dadas por parámetro. En la tabla 4, se puede percibir los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos ver que el valor máximo al que se puede llegar es de 1,49, siendo este el peor modelo entrenado para el lenguaje de Wayuunaiki.

Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1	1.3933	1.5107	0.8563	18.0712
2	1.598	1.3444	0.9626	18.0648
3	1.4277	1.2551	1.1025	17.9695
4	1.4152	1.2	1.1361	17.9426
5	1.1671	1.1565	1.2243	17.8416
6	1.1777	1.1217	1.2874	17.8809
7	1.4485	1.0955	1.3318	17.9663
8	1.3209	1.0729	1.3889	17.967
9	1.394	1.0557	1.4082	17.8646
10	1.0608	1.0435	1.4463	17.9294
11	1.0713	1.0323	1.4558	17.9015
12	0.976	1.0248	1.4666	17.9103
13	1.0782	1.0191	1.484	17.8929
14	1.045	1.015	1.4869	17.8875
15	0.9936	1.0136	1.4957	17.8854

**Cuadro 4** Resultados modelo T5 base para Wayuunaiki

Adicionalmente, se puede observar en las imágenes 8, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 1,08 para los datos de entrenamiento y del 1,01, para los datos de validación.



**Figura 8** Resultados función loss T5 Wayuunaiki. a) Datos de entrenamiento, b) Datos de validación

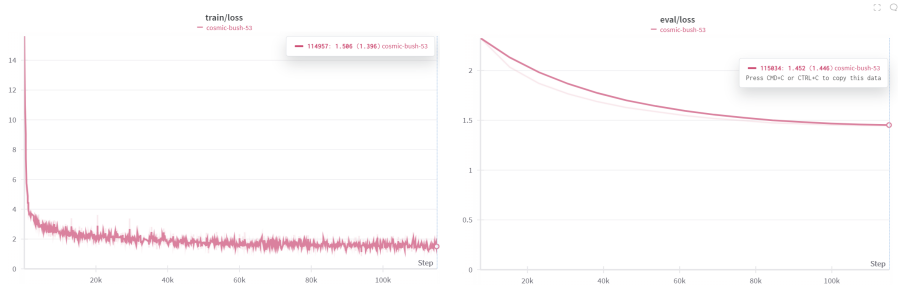
### 3.1.4. MT5

El modelo fue entrenado por 11h 52m 48s, y no paró debido al earlystopping, sino completó las 15 épocas de entrenamiento inicialmente dadas por parámetro. En la tabla 5, se puede observar los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos observar que el valor máximo al que se puede llegar es de 3,08, sacrificando un poco del training loss, no obstante, se puede observar que el loss no disminuye al nivel de los otros modelos, por lo tanto, este modelo pudo haber sido entrenado por un par de épocas adicional.

Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1	2.351500	2.325468	1.209200	16.887800
2	2.322900	2.034342	1.394800	16.696700
3	1.828000	1.870699	1.794800	16.729000
4	1.769200	1.765399	1.981600	16.553800
5	3.107300	1.687816	2.233500	16.666200
6	1.872600	1.627995	2.423100	16.566000
7	1.861100	1.587044	2.583900	16.572400
8	1.477700	1.547600	2.741500	16.565100
9	1.533100	1.518059	2.810500	16.586200
10	1.634800	1.497183	2.938600	16.582000
11	1.219500	1.473864	2.933900	16.578000
12	1.865100	1.464697	3.005800	16.543100
13	1.348400	1.453717	3.079000	16.551300
14	1.567500	1.448385	3.092500	16.549600
15	1.651800	1.446463	3.088000	16.546800

**Cuadro 5** Resultados modelo MT5 para Wayuunaiki

Adicionalmente, se puede observar en las imágenes 9, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 1,5 para los datos de entrenamiento y del 1,4, para los datos de validación.



**Figura 9** Resultados función loss MT5 Wayuunaiki. a) Datos de entrenamiento, b) Datos de validación

## 3.2. Resultados Nasa Yuwe

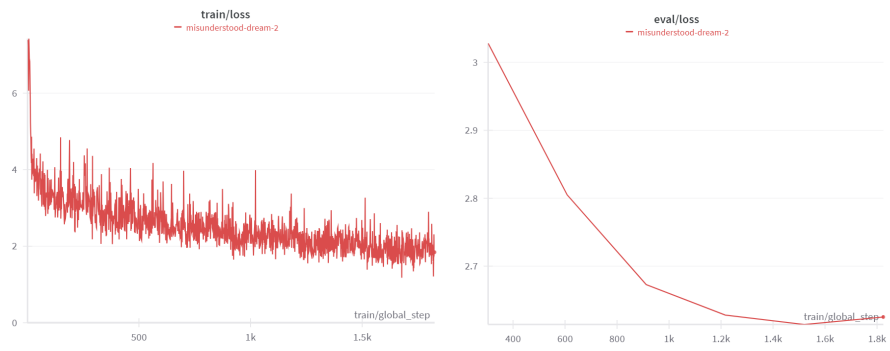
### 3.2.1. Bart

El modelo se entrenó por 32m 23s e inicialmente se propuso para que fuera entrenado por 15 épocas, sin embargo, el modelo paró el entrenamiento producto del early stopping en la época número 6, por lo que se puede mostrar que Bart es una arquitectura más sencilla y no pudo mejorar mucho más. En la tabla 6, se puede observar los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos observar que el valor máximo al que se puede llegar es de 1,5675.

Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1	2.6025	3.0281	0.0	7.7339
2	3.694	2.8050	0.0	5.3307
3	2.3214	2.6729	0.0	11.5929
4	2.0	2.6280	0.4389	10.8669
5	2.0676	2.6142	1.5675	9.6904
6	1.8422	2.6252	0.233	11.0184

**Cuadro 6** Resultados modelo Bart para Nasa Yuwe

Adicionalmente, se puede observar en las imágenes 10, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 1,80 para los datos de entrenamiento y del 2,62, para los datos de validación.



**Figura 10** Resultados función loss Bart Nasa Yuwe. a) Datos de entrenamiento, b) Datos de validación

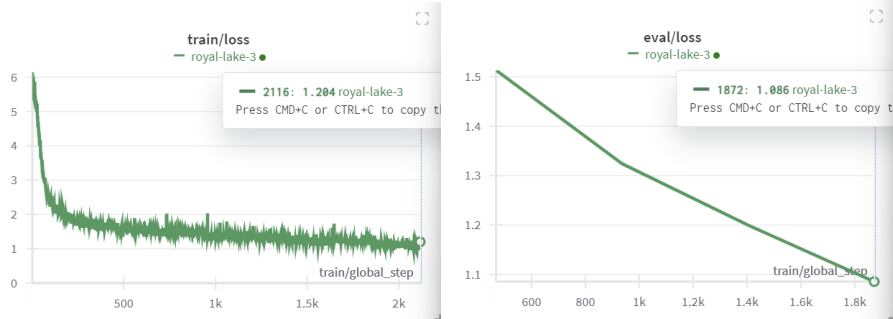
### 3.2.2. Llama2

El modelo se entrenó por 9h 2m 47s, el modelo se entrenó por 15 épocas sin parar antes en el early stopping. En la tabla 7, se puede observar los resultados por cada una de las épocas del training y validation loss. Para la métrica de BLEU se calculó el promedio para este modelo, obteniendo 0,02208.

Epochs	Training Loss	Validation Loss	Bleu
1	1.526	1.513408	-
2	1.318	1.324283	-
3	1.1914	1.199728	-
4	1.1078	1.085581	-
5	0.996	0.983942	-
6	0.8253	0.883787	-
7	0.7293	0.800595	-
8	0.7336	0.708206	-
9	0.6138	0.637108	-
10	0.4983	0.566668	-
11	0.4381	0.507155	-
12	0.4317	0.4584	-
13	0.3372	0.42045	-
14	0.3267	0.393277	-
15	0.3658	0.38098	-

**Cuadro 7** Resultados modelo Llama2 para Nasa Yuwe

En las gráficas que se pueden observar en las imágenes 11, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 1,20 para los datos de entrenamiento y del 1,86, para los datos de validación.

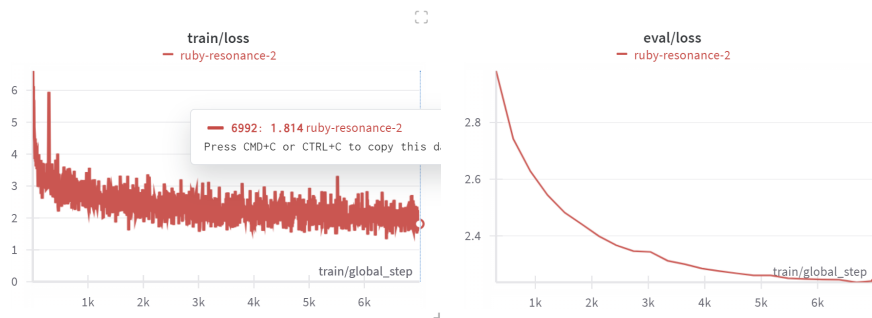


**Figura 11** Resultados función loss Llama Nasa Yuwe. a) Datos de entrenamiento, b) Datos de validación

### 3.2.3. T5 base

Este modelo se entrenó por 31m 19s. Para correrlo, se ingresó como hiperparámetro 30 épocas, sin embargo, se puede observar que paró debido al earlystopping en la época 23. En la tabla 8, se puede percibir los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos ver que el valor máximo al que se puede llegar es de 1,0985, sin embargo, también se puede observar que en 5 épocas el BLEU es de 0,0.

Adicionalmente, se puede observar en las imágenes 12, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 1,81 para los datos de entrenamiento y del 2,24, para los datos de validación.



**Figura 12** Resultados función loss T5 Nasa Yuwe. a) Datos de entrenamiento, b) Datos de validación

Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1	2.6692	2.9825	0.8944	6.2582
2	2.6593	2.7422	0.0	6.9895
3	2.5452	2.6276	0.0	7.1924
4	2.5998	2.5437	0.0	7.3347
5	3.0987	2.4819	0.0	7.5204
6	2.3259	2.4409	0.0	7.4466
7	3.2006	2.3988	0.6694	7.4058
8	1.989	2.3669	0.6097	8.1383
9	2.3702	2.3464	0.9537	8.1542
10	2.3841	2.3434	0.9045	7.7852
11	2.2193	2.3119	0.9082	8.22
12	1.8003	2.2848	1.0315	8.2055
13	1.8003	2.2848	1.0315	8.2055
14	1.9862	2.2756	0.6622	8.2134
15	2.3814	2.2678	0.6688	8.1634
16	2.145	2.2606	0.8214	8.2754
17	2.1513	2.2605	1.0985	8.2635
18	2.249	2.2506	1.0695	8.1726
19	2.3972	2.2477	0.663	8.22
20	2.1375	2.2458	0.612	8.1515
21	2.4343	2.2451	0.6825	8.1871
22	2.9682	2.2361	0.6095	8.2306
23	1.8138	2.2411	0.608	8.108

**Cuadro 8** Resultados modelo T5 base para Nasa Yuwe

#### 3.2.4. MT5

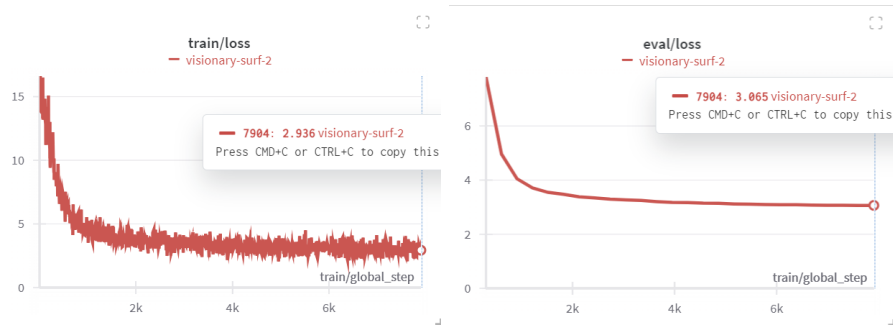
Este modelo se entrenó por 46m 34s. Para correrlo, se ingresó como hiperparámetro 30 épocas, sin embargo, se puede observar que paró debido al earlystopping en la época 26. En la tabla 9, se puede percibir los resultados por cada una de las épocas del training y validation loss, y la métrica de BLEU. Podemos ver que el valor máximo al que se puede llegar es de 0,5194, siendo este el peor modelo para la lengua Nasa Yuwe.

Adicionalmente, se puede observar en las imágenes 13, cómo se ve la curva de la función loss tanto para los datos de entrenamiento, como de validación. Se puede observar que en ambos casos se estabiliza, al rededor del 2,93 para los datos de entrenamiento y del 3,06, para los datos de validación.



Epoch	Training Loss	Validation Loss	Bleu	Gen Len
1.0	9.0597	7.8135	0.0148	4.6469
2.0	6.2294	4.9617	0.0	3.9209
3.0	4.8326	4.0494	0.0	4.3808
4.0	4.582	3.7069	0.0	5.3979
5.0	5.4762	3.5463	0.0	5.6759
6.0	4.3875	3.4731	0.0	5.6258
7.0	4.2873	3.3832	0.0	5.5455
8.0	4.1326	3.3424	0.0	5.4756
9.0	3.5728	3.2956	0.0	5.1792
10.0	3.1873	3.2690	0.0	5.5903
11.0	2.9436	3.2465	0.1237	5.7655
12.0	4.2955	3.2054	0.1741	5.4466
13.0	3.8722	3.1764	0.1887	5.2161
14.0	3.5391	3.1688	0.0951	5.7312
15.0	3.8012	3.1480	0.1948	5.2964
16.0	3.1148	3.1401	0.2397	5.7589
17.0	3.2699	3.1186	0.33	5.386
18.0	4.3355	3.1092	0.4637	5.1383
19.0	3.5792	3.0966	0.3286	5.4374
20.0	3.1429	3.0923	0.418	5.2964
21.0	3.4155	3.0900	0.3938	5.4848
22.0	3.4515	3.0755	0.4062	5.4124
23.0	2.8244	3.0717	0.4218	5.3663
24.0	2.9253	3.0663	0.3633	5.5692
25.0	2.1757	3.0640	0.4768	5.4282
26.0	2.9356	3.0646	0.5194	5.3808

**Cuadro 9** Resultados modelo MT5 para Nasa Yuwe

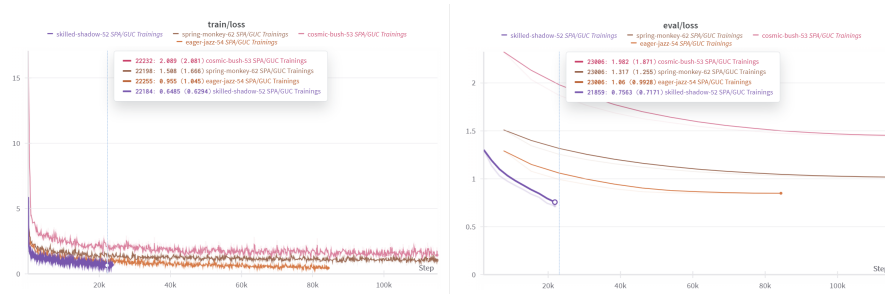


**Figura 13** Resultados función loss MT5 Nasa Yuwe. a) Datos de entrenamiento, b) Datos de validación

### 3.3. Resultados generales

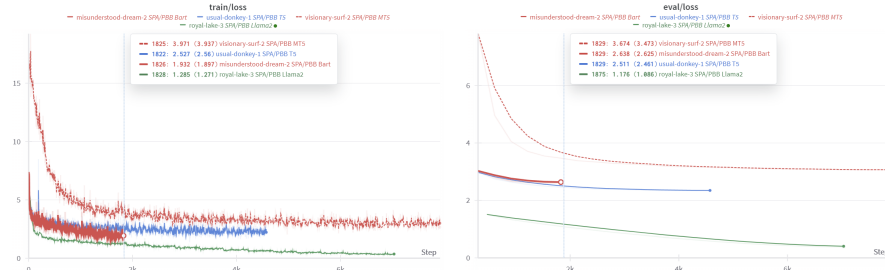
Ahora bien, como resultados generales, se puede observar que los modelos tienen mejores resultados para el lenguaje Wayuunaiki que para Nasa Yuwe. Esto se puede llegar a deber a la baja cantidad de datos para el conjunto de la lengua Nasa Yuwe, que como se menciona en la sección 2.1.2, se tiene únicamente, 3794 pares de registros, a diferencia de la lengua Wayuunaiki (sección 2.1.1) la cual contiene, 119808 pares de registros.

En las gráficas 14 se pueden observar la comparación de cada uno de los modelos para el lenguaje de Wayuunaiki. La gráfica en morado representa el modelo de Llama, la gráfica en café el modelo T5, la rosada el modelo MT5 y la naranja el modelo de Bart.



**Figura 14** Resultados función loss para Wayuunaiki. a) Datos de entrenamiento, b) Datos de validación

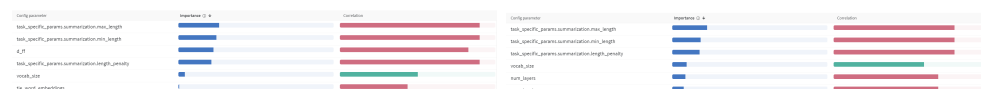
Por otro lado, en las gráficas 15 se pueden observar la comparación de cada uno de los modelos para el lenguaje de Nasa Yuwe. La gráfica en verde representa el modelo de Llama, la gráfica en azul el modelo T5, la roja punteado el modelo MT5 y la roja continuo el modelo de Bart.



**Figura 15** Resultados función loss para Nasa Yuwe. a) Datos de entrenamiento, b) Datos de validación

También podemos observar en la imagen 16 la importancia de los parámetros para cada uno de los modelos de los diferentes lenguajes, y podemos observar que para

ambos modelos son importantes los mismos dos primeros parámetros, sin embargo, la importancia va variando dependiendo del lenguaje.



**Figura 16** Importancia de Parametros a) Wayuuunaiki, b) Nasa Yuwe

Adicionalmente, en la tabla 10 hacemos una comparación del mejor BLEU obtenido por modelo para cada lenguaje.

Modelo	Wayuunaiki	Nasa Yuwe
Bart	3.2289	1.5675
Llama2	0.1347	0.0220
T5 base	1.4957	1.0985
MT5	3.0925	0.5194

**Cuadro 10** Comparación de los modelos para cada lenguaje

Como se puede evidenciar en la tabla 10, el mejor modelo para las dos lenguas es Bart, y este a pesar de que es un modelo sencillo, funciona bastante bien, y tiene buenos resultados.

### 3.4. Comparación con trabajos relacionados

Para esto se va a comparar los resultados en métricas BLEU del mejor modelo por lenguaje con los resultados dados en los trabajos *Enriching Wayunaiki-Spanish Neural Machine Translation with Linguistic Information* [2], que para esta comparación se nombrará como EWS y *Machine Translation Strategies for Low-Resource Colombian Indigenous Languages* [3], que para esta comparación se mencionará como MTS. Esto teniendo en cuenta que el trabajo EWS [2] únicamente hace uso del lenguaje Wayunaiki, a diferencia del trabajo MTS [3] en el cual se hace uso de los dos lenguajes usados en el presente.

Como se puede observar en la tabla 11, los mejores resultados dados para el lenguaje Wayuunaiki fueron del presente trabajo, sin embargo, la comparación no es necesariamente justa teniendo en cuenta que los modelos están entrenados con contextos diferentes, puesto que para el caso de EWS, los resultados dados son para un contexto religioso, siendo este el mejor resultado dado para dicho trabajo a diferencia del presente trabajo el cual tiene un contexto general de la lengua. Con respecto al trabajo MTS, este es el mejor promedio dado para distintas variaciones del modelo

Trabajo	Wayuunaiki	Nasa Yuwe
Presente	3.2289	1.5675
EWS	$1.2 \pm 0.3$	-
MTS	2.566	2.055

**Cuadro 11** Comparación de resultados con trabajos relacionados

implementado e igualmente se tiene en cuenta que el dataset usado tiene menos pares de sentencias, puesto que tiene 9081.

Igualmente, se puede observar que a diferencia del lenguaje Wayuunaiki, los mejores resultados para el lenguaje Nasa Yuwe son dados por el trabajo MTS en donde una de las razones es la cantidad de pares de sentencias usadas en el dataset, que para ello en dicho trabajo se usó 7923.

## 4. Conclusión

Teniendo en cuenta la comparación de los resultados dados en el presente trabajo frente a los resultados dados por trabajos relacionados e igualmente la comparación entre lenguajes, se puede concluir que uno de los factores más importantes, si no es él más, es la cantidad de datos disponibles para realizar los modelos. Esto se evidencia directamente al comparar los resultados nuestros con los del trabajo MTS, en el cual se tiene menor cantidad de datos para el lenguaje Wayuunaiki, pero mayor cantidad de datos para el Nasa Yuwe y es de ahí que los resultados de BLEU dado para Wayuunaiki es peor y para Nasa Yuwe es mejor a pesar de ser el mismo modelo. Igualmente, gracias a la comparación que se tiene con el trabajo EWS, se puede evidenciar la importancia del contexto de los datos, al tener un dataset con un contexto más amplio, como es nuestro caso, se logran obtener mejores resultados.

Dada la cantidad de datos en el conjunto seleccionado para Wayuu, el proceso de entrenamiento toma un tiempo significativo, con lo que se aprecia que si bien el mejor resultado fue con BART, los resultados con llama 2 podrían mejorar si se entrena por mas tiempo, sin embargo, considerando el costo y la complejidad computacional, un modelo ajustado para la tarea de traducción funciona mejor que un modelo de lenguaje general ajustado para la misma tarea.

Por otro lado, teniendo en cuenta los modelos implementados y los resultados dados para cada uno de estos en cada lenguaje, se evidencia que no necesariamente una arquitectura más compleja da mejores resultados en la tarea de traducción de lenguas indígenas como el Wayuunaiki y el Nasa Yuwe.

## Referencias

- [1] UNESCO: Multilingüismo Y Diversidad Lingüística

- [2] Nora Graichen, J.V.G., España-bonet, C.: Enriching wayúunaiki-spanish neural machine translation with linguistic information
- [3] Cárdenas, I.D.S.: Machine Translation Strategies for Low-Resource Colombian Indigenous Languages
- [4] Mapa Sonoro - Lenguas Nativas de Colombia
- [5] Tatoeba-Challenge
- [6] Amaya, R.J.N.: Spanish-wayuunaki
- [7] OLAC Resources in and About the Wayuu Language
- [8] Wayuunaiki (guajiro). Diccionario Español
- [9] Constitucion Politica de Colombia en Nasa Yuwe
- [10] Diccionario Nasa (Páez)- Español
- [11] OLAC Resources in and About the Páez Language
- [12] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension (2019)
- [13] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models (2023)
- [14] Roberts, A., Raffel, C., Lee, K., Matena, M., Shazeer, N., Liu, P.J., Narang, S., Li, W., Zhou, Y.: Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google (2019)
- [15] Alammam, J.: The Illustrated Transformer
- [16] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer (2021)
- [17] Kishore Papineni, T.W. Salim Roukos, Zhu, W.-J.: Bleu: A method for automatic evaluation of machine translation (2002)