TRADUCCIÓN AUTOMÁTICA Y CONTINUA DE LENGUA DE SEÑAS UTILIZANDO REPRESENTACIONES INTERMEDIAS BASADAS EN GLOSAS

FREDY ALEJANDRO MENDOZA LÓPEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2022

TRADUCCIÓN AUTOMÁTICA Y CONTINUA DE LENGUA DE SEÑAS UTILIZANDO REPRESENTACIONES INTERMEDIAS BASADAS EN GLOSAS

FREDY ALEJANDRO MENDOZA LÓPEZ

Trabajo de grado para optar al título de
Ingeniero de Sistemas

Director:
Fabio Martínez Carrillo
Doctor en Ingeniería de Sistemas y Computación

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2022

DEDICATORIA

Para mi mamá Sandra, mi papá Fredy
y mi hermana Paula.
Ellos son el motor de mi vida.

AGRADECIMIENTOS

El autor expresa su agradecimiento:

A mi director, el doctor Fabio Martínez, un gran profesional que me acompañó y orientó de forma excepcional en el desarrollo de este trabajo. Asimismo, al magister Jefferson Rodríguez, quien participó activamente en todas las fases del proyecto. A ambos les agradezco la paciencia, constancia y ayuda, la cual se ve reflejada en el presente trabajo.

Al grupo de investigación BIVL^{2}ab, el cual me brindó las herramientas necesarias para el desarrollo de mi trabajo y me dio la oportunidad de conocer excelentes personas.

A la Universidad Industrial de Santander, alma máter que me permitió formarme académicamente como profesional.

A mis padres, Sandra y Fredy, quienes me han acompañado durante toda mi vida y me han formado en la persona que soy, siendo ellos una pieza fundamental de motivación y felicidad en mi día a día.

A mi hermana, Paula, la cual ha crecido conmigo y me ha mostrado su apoyo incondicional.

A mi familia, en especial a mis abuelos, Aminta, Isidora, Nicanor y Manuel, mi hermana, Karol y mis primos, Andrés, Jeysson, Jhon, Daniela y Camilo.

A mis amigos, Brayan, Jenny, Santiago, Orlando, Jennifer, Jessica, Juan Felipe, Andrés Felipe y David, quienes me acompañaron a lo largo de mi carrera académica y me seguirán acompañando en mi vida personal.

CONTENIDO

LISTA DE FIGURAS

pág.

LISTA DE CUADROS

pág.

RESUMEN

TÍTULO: TRADUCCIÓN AUTOMÁTICA Y CONTINUA DE LENGUA DE SEÑAS UTILIZANDO REPRESENTACIONES INTERMEDIAS BASADAS EN GLOSAS [*]

AUTOR: FREDY ALEJANDRO MENDOZA LÓPEZ [**]

PALABRAS CLAVE: TRADUCCIÓN CONTINUA DE LENGUA DE SEÑAS, REPRESENTACIÓN CINEMÁTICA, GLOSAS, ANÁLISIS DE VIDEO, *TRANSFORMER*, REPRESENTACIONES DE APRENDIZAJE PROFUNDO.

DESCRIPCIÓN: La ausencia de una comunicación efectiva con la población sorda representa la principal brecha social en esta comunidad. Además, la lengua de señas, que constituye la principal herramienta de comunicación de los sordos, es ágrafa, es decir, no existe una representación escrita. En consecuencia, uno de los principales retos actuales es la traducción automática entre la representación espaciotemporal de los signos y el lenguaje de texto natural. En el estado de arte, enfoques recientes se basan en arquitecturas codificador-decodificador, donde las estrategias más relevantes integran módulos de atención para mejorar las correspondencias no lineales, sin embargo, siguen estando limitadas por la información redundante de las secuencias de video. Además, muchas de estas aproximaciones requieren complejos esquemas de entrenamiento y arquitectura para lograr predicciones razonables, debido a la ausencia de proyecciones de texto intermedias. Las glosas son proyecciones escritas nativas de un símbolo semántico, expresado a partir de un conjunto de señas, que pueden ser clave como representación intermedia para lograr traducciones coherentes. Este trabajo introduce una arquitectura *transformer* multitarea que incluye una representación de aprendizaje de glosas para lograr una traducción más adecuada. El enfoque propuesto incluye una representación de movimiento densa que exalta los gestos e incluye información cinemática, un componente clave en la lengua de señas. A partir de esta representación es posible evitar información de fondo y explotar la geometría de las señas, además, incluye representaciones espaciotemporales que facilitan el alineamiento entre los gestos y las glosas como representación textual intermedia. El enfoque propuesto supera las estrategias evaluadas en el estado del arte en el conjunto de datos CoL-SLTD, logrando un BLEU-4 de $72,64\,\%$ en la división 1 y un BLEU-4 de $14,64\,\%$ en la división 2. Además, la estrategia fue validada en el conjunto de datos RWTH-PHOENIX-Weather 2014 T, logrando un notable BLEU-4 de $11,58\,\%$.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Fabio Martínez Carrillo. Doctor en Ingeniería de Sistemas y Computación.

ABSTRACT

TITLE: AUTOMATIC AND CONTINUOUS SIGN LANGUAGE TRANSLATION USING AN INTERMEDIATE REPRESENTATION BASED ON GLOSSES [*]

AUTHOR: FREDY ALEJANDRO MENDOZA LÓPEZ [**]

KEYWORDS: CONTINUOUS SIGN LANGUAGE TRANSLATION, KINEMATIC REPRESENTATION, GLOSSES, VIDEO ANALYSIS, TRANSFORMER, DEEP LEARNING REPRESENTATIONS.

DESCRIPTION: The absence of an effective communication with deaf population represents the main social gap with this community. Furthermore, the sign language, the main deaf communication tool, is unlettered, *i.e.,* there is not a written representation. In consequence, a main challenge today is the automatic translation among spatiotemporal sign representation and natural text language. In the state-of-the-art, recent approaches are based on encoder-decoder architectures, where the most relevant strategies integrate attention modules to enhance non-linear correspondences, however, they are still limited by the redundant background information of the video-sequences. Besides, much of these approximations requires complex training and architectural schemes to achieve reasonable predictions, because the absence intermediate text projections. The glosses are native written projections of a semantic symbol, expressed from a set of signs, that might be key as intermediate representation to achieve coherent translations. This work introduces a multitask transformer architecture that includes a gloss learning representation to achieve a more suitable translation. The proposed approach includes a dense motion representation that enhance gestures and includes kinematic information, a key component in sign language. From this representation it is possible to avoid a background information and exploit the geometry of the signs, in adittion, it includes spatiotemporal representations that facilitate the alignment between gestures and glosses as an intermediate textual representation. The proposed approach outperforms the state-of-the-art evaluated on the CoL-SLTD dataset, achieving a BLEU-4 of $72,64\,\%$ in split 1, and a BLEU-4 of $14,64\,\%$ in split 2. Furthermore, the strategy was validated on the RWTH-PHOENIX-Weather 2014 T dataset, achieving a remarkable BLEU-4 of $11,58\,\%$.

---

[*] Bachelor Thesis

[**] Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering. Advisor: Fabio Martínez Carrillo. Ph.D. in Computer and Systems Engineering.

INTRODUCCIÓN

Hoy en día se estima que aproximadamente 1.500 millones de personas poseen algún grado de pérdida auditiva en todo el mundo [1]. La lengua de señas (LS) es el principal mecanismo de la comunicación con personas sordas o con dificultades auditivas. Esta lengua se compone de movimientos y expresiones gesto-visoespaciales, donde coexisten complejas interacciones manuales (uso de las manos) y no manuales (uso de la boca, ojos, rostro y cuerpo). Como cualquier lengua, existe una riqueza gramatical intrínseca, que en este caso se refleja con múltiples variaciones gestuales y expresivas. Estos aspectos hacen que el modelado de las LS sea una tarea difícil, incluso para las metodologías más avanzadas de visión por computador y aprendizaje profundo. De hecho, hoy en día existen más de 150 LS oficiales con múltiples variaciones en cada país [2], conllevando a diferentes gestos, expresiones y reglas gramaticales en cada una de ellas. Por ejemplo, Colombia cuenta con la LS colombiana, pero en cada región existen variaciones propias a la cultura e idiosincrasia de cada región. Además, la LS se caracteriza por ser ágrafa, es decir, no tienen una representación escrita directa, lo que dificulta la estructuración y generalización del lenguaje, implicando grandes retos para encontrar correspondencia con otros lenguajes textuales. En este caso, la correspondencia con el medio escrito se ha logrado a partir de las glosas. Específicamente, una glosa hace referencia a una proyección escrita, la cual corresponde a la palabra que mejor representa la información que se desea transmitir en una única o múltiples señas.

En la actualidad, los enfoques del estado del arte han avanzado en la traducción au-

---

[1] World Health Organization y col. “World report on hearing”. En: (2021), pág. 10.

[2] David M. Eberhard, Gary F. Simons y Charles D. Fennig. *Ethnologue: Languages of the World*. 2022. URL: https://www.ethnologue.com/subgroups/sign-language (visitado 03-06-2022).

tomática de la LS basándose en arquitecturas codificador-decodificador, las cuales, incluyen niveles recurrentes para modelar secuencias de video a texto [3]. Estos enfoques han reportado resultados destacables en muestras con múltiples variaciones en conjuntos de datos relativamente grandes. Sin embargo, estas representaciones profundas se limitan a capturar principalmente pequeñas dependencias temporales, dejando de lado las relaciones de mayor temporalidad que son claves en la LS. Recientemente, los enfoques *transformer*, basados en un modelo totalmente fundamentado en módulos de atención [4], se han aproximado a la tarea de traducción computando matrices atencionales que recuperan interacciones complejas y de larga duración entre las señas, siendo más robustas para la representación del lenguaje [5][6]. Estas arquitecturas suelen optimizarse sobre representaciones convolucionales 2D de videos sin procesar, restando importancia al procesamiento temporal del movimiento intrínseco en los gestos, el cual, ha demostrado ser relevante en la representación del lenguaje [7].

A pesar de los notables avances en estos esquemas computacionales, la mayoría de las propuestas realizan una proyección y correspondencia directa de los componentes gesto-visuales con la comunicación escrita, dejando de lado la representación

---

[3] Necati Cihan Camgoz y col. "Neural sign language translation". En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, págs. 7784-7793.

[4] Ashish Vaswani y col. "Attention is all you need". En: *Advances in neural information processing systems*. 2017, págs. 5998-6008.

[5] Necati Cihan Camgoz y col. "Sign language transformers: Joint end-to-end sign language recognition and translation". En: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, págs. 10023-10033.

[6] Kayo Yin y Jesse Read. "Better sign language translation with STMC-transformer". En: *arXiv preprint arXiv:2004.00588* (2020).

[7] Jefferson Rodriguez y col. "Understanding Motion in Sign Language: A New Structured Translation Dataset". En: *Proceedings of the Asian Conference on Computer Vision*. 2020.

en glosas. Esto, debido a la alta asincronía entre los dos dominios, resulta en modelos con relaciones altamente complejas y poco eficientes entre las dos lenguas. De hecho, incluso en el análisis lingüístico llevado por expertos, el uso de glosas resulta imprescindible para una correspondencia ágil y efectiva entre los dos canales de comunicación [8]. Trabajos preliminares en el estado del arte han mostrado que la inclusión de una representación intermedia puede mejorar la representación de la LS [5].

Este trabajo introduce una arquitectura *transformer* multitarea que incluye una representación intermedia de glosas para lograr una traducción más adecuada. El enfoque propuesto también incluye una representación de movimiento denso que realza los gestos del fondo e incluye información cinemática, un componente clave en la LS, permitiendo discriminar entre los gestos realizados por el intérprete y la escena de captura. Estas primitivas cinemáticas se obtienen a partir de campos densos de movimiento aparente del componente gestual de las frases de LS. Asimismo, estas representaciones densas capturan directamente toda la dinámica gestual, pero también conservan la forma de los signos durante las secuencias. El modelo *transformer* recibe descriptores de movimiento de video para realizar una optimización conjunta *end-to-end* (entrenamiento extremo a extremo en la arquitectura), asociando inicialmente los descriptores cinemáticos extraídos con las representaciones de glosa en un paso intermedio, con el fin de generar una traducción textual final más alineada y precisa. La metodología propuesta fue evaluada en dos conjuntos de datos públicos: el conjunto CoL-SLTD y el conjunto RWTH-PHOENIX-Weather 2014 T.

---

[8] Samuel J Supalla, Jody H Cripps y Andrew PJ Byrne. “Why American sign language gloss must matter”. En: *American annals of the deaf* 161.5 (2017), págs. 540-551.

1. MARCO TEÓRICO Y TRABAJOS PREVIOS

1.1. LAS GLOSAS Y LA LENGUA DE SEÑAS

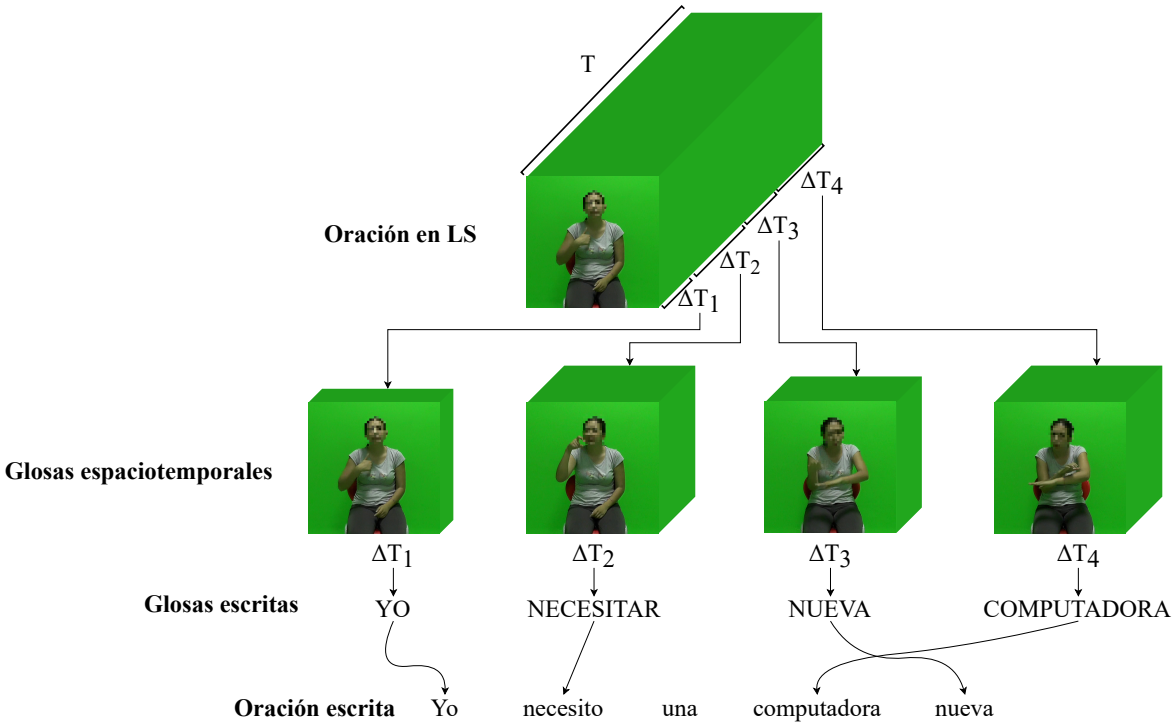
La LS es el principal soporte de comunicación en la población sorda. Los componentes básicos de esta lengua se basan en expresiones gesto-visoespaciales que representan un fragmento de información. En cada componente se involucran procesos complejos donde actúan interacciones manuales y no manuales para definir la comunicación. Estos gestos están acompañados de movimientos particulares que enriquecen esta lengua y sus múltiples representaciones.

La LS se caracteriza por ser ágrafa, es decir, no existe una traducción directa en algún idioma escrito. Particularmente, las glosas son la proyección directa de la LS en símbolos escritos. Esta representación es el soporte escrito que mejor enmarca el conjunto de características lingüísticas de la LS. Una glosa es una proyección textual que coincide con la palabra que describe con mayor precisión la información que se transmite en una o varias señas. Estas glosas se escriben en mayúscula, con el fin de diferenciarlas de las oraciones en lenguajes escritos [9]. A partir de la representación en glosas, es posible obtener una traducción a un idioma escrito. No obstante, existe una alta asincronía a la hora de realizar traducciones de glosas a lengua escrita, debido a que la correspondencia entre los dos dominios no es lineal, varían en longitud y existen importantes diferencias en las reglas gramaticales. La figura 1 muestra un ejemplo de traducción de LS a lenguaje escrito.

---

[9] María Ignacia Massone. “El habla visual: lingüística de las lenguas de señas”. En: *Signo y seña* 2 (1993), págs. 18-27.

Figura 1. Ejemplo de traducción de LS a lenguaje escrito utilizando glosas. Una frase de LS se puede fraccionar en glosas a través de su dimensión temporal, donde una glosa corresponde a un lapso de tiempo ΔT. Una vez obtenida la oración a nivel de glosas, se realiza la traducción a texto escrito.

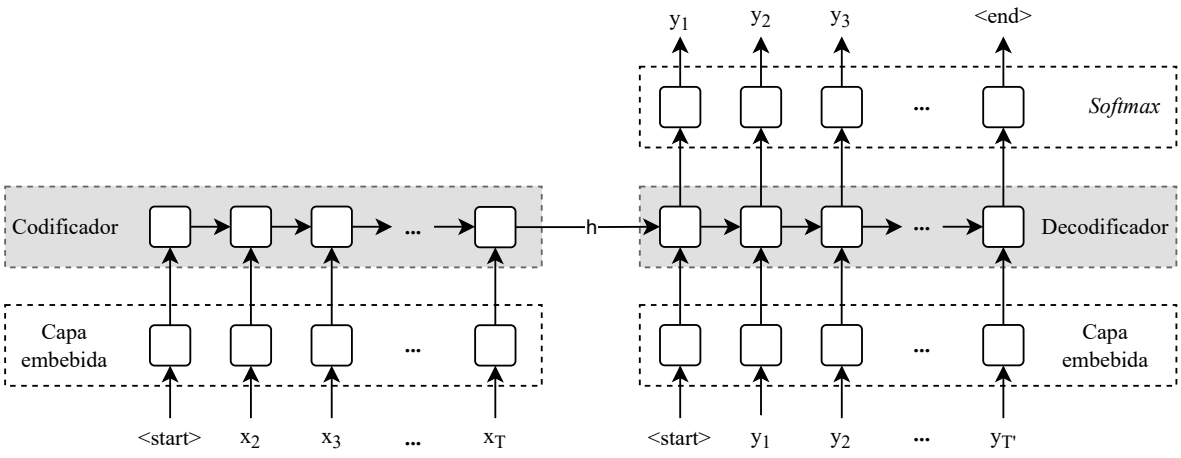


1.2. REPRESENTACIONES CODIFICADOR-DECODIFICADOR

La sincronización y el uso de diferentes modalidades de captura que registran temporalmente variables (secuencias) con diversos componentes, dimensiones y resoluciones temporales de registro, son unos de los principales desafíos en el procesamiento natural del lenguaje y la visión por computador. Los modelos de tipo codificador-decodificador afrontan este problema de forma sobresaliente, según resultados evidenciados en el estado del arte. Particularmente, una secuencia $\mathbf{X} = (x_1, \cdots, x_T)$, donde T hace referencia a la longitud temporal, puede representar un modo de captura de información (texto, video, audio, entre otros) descrita en un espacio n-dimensional. Por ejemplo, en las aplicaciones de traducción de la

LS se busca que una secuencia de video que registra un conjunto de fotogramas $\mathbf{V} = (v_1, \cdots, v_T)$, se correlacione con respecto a los parámetros semánticos de una secuencia de texto $\mathbf{W} = (w_1, \cdots, w_M)$ que se ordena de forma coherente y secuencial.

Figura 2. Modelo codificador-decodificador. El modelo transforma una secuencia $\mathbf{X} = (x_1, \cdots, x_T)$ en una secuencia $\mathbf{Y} = (y_1, \cdots, y_{T'})$, donde T y T' pueden representar diferentes longitudes.



Entonces, como se ilustra en la figura 2, la secuencia de video $\mathbf{V}$ es proyectada a la representación codificador-decodificador, donde inicialmente se obtienen vectores embebidos o descriptores que permiten representar la información en baja dimensionalidad, resaltando los patrones más relacionados con la tarea. Estos vectores embebidos son típicamente tratados en capas densas o recurrentes, con el fin de explotar patrones temporales de las secuencias donde se puedan incluir variantes de estas, buscando obtener vectores que resuman la información. Desde el codificador, estos embebidos son las unidades de información que se correlacionan con el decodificador para lograr un emparejamiento y asociación con las secuencias de texto. Desde el decodificador, la secuencia textual que tiene correspondencia con el video también es proyectada a una capa embebida con el fin de obtener una re-

presentación compacta. Luego, estos vectores se dan como entrada a un esquema decodificador, el cual no solamente explota la temporalidad de la secuencia de texto, sino que introduce diferentes mecanismos para integrar las unidades resultantes del codificador, logrando de manera conjunta una estimación de la siguiente palabra más probable. Estos esquemas de codificador-decodificador tienen múltiples variantes en sus diferentes capas de procesamiento y pueden incluir nuevos mecanismos de relación entre las secuencias.

1.2.1. Arquitectura *transformer*. Los modelos *transformer* son hoy en día una de las representaciones codificador-decodificador más efectivas en el problema de la traducción de la LS. Específicamente, la arquitectura *transformer* se basa netamente en módulos de atención para capturar información temporal [4]. Este enfoque supone una importante ventaja, debido a que, al no utilizar una estrategia recurrente, se elimina la limitación de memoria al modelar secuencias de gran tamaño temporal. En términos simples, el codificador recibe como entrada una secuencia $\mathbf{X}$ y tiene como función encontrar relaciones temporales entre cada elemento de este conjunto de entrada, donde estas relaciones se representan en un espacio latente $\mathbf{Z} = (z_1, \cdots, z_T)$. Por otro lado, el decodificador recibe dos entradas, el conjunto $\mathbf{Z}$ y la secuencia $\mathbf{Y}$. Con base en esto, su función principal se basa en dos objetivos, encontrar relaciones temporales entre cada elemento de $\mathbf{Y}$ y determinar proyecciones entre la secuencia de entrada y salida. Ambos módulos están compuestos por una secuencia B de capas idénticas.

En la literatura, los modelos recurrentes codificador-decodificador han permitido abordar problemas que involucran la proyección entre diversas representaciones temporales. A pesar de esto, su modelamiento se ve justamente restringido por la comunicación típica entre los dos módulos a través de un único vector embebido que representa la secuencia de entrada, pero por su naturaleza recurrente pondera con mayor relevancia la información más reciente. Este efecto, conocido comúnmen-

te como “cuello de botella”, reduce la posibilidad de explorar relaciones altamente no lineales y altas correspondencias semánticas espaciadas de forma significativa en la secuencia (consecuencia inherente al carácter del lenguaje). Al no utilizar una estrategia recurrente, el modelo *transformer* se basa en módulos de atención para extraer características temporales, donde se explotan las principales relaciones entre las secuencias. Esta arquitectura emplea una atención de tipo *multi-head*, la cual está conformada por un conjunto de módulos de atención de tipo *scaled dot-product*. A continuación, se explica de forma específica el funcionamiento de cada uno de estos enfoques de atención.

1.2.1.1. Modelo de atención *scaled dot-product*. Este tipo de atención se basa en operaciones vectoriales utilizando el producto punto. El algoritmo recibe como entrada tres vectores: key ($\mathbf{K}$), query ($\mathbf{Q}$) y value ($\mathbf{V}$). Estos vectores en las aplicaciones de traducción corresponden a los embebidos de las secuencias de video o de texto. Los vectores $\mathbf{K}$ y $\mathbf{Q}$ tienen dimensión $\mathbf{d_k}$, mientras que el vector $\mathbf{V}$ es de dimensión $\mathbf{d_v}$. La matriz de atención se computa de la forma: $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$.

Al utilizar el producto punto, el resultado obtenido entre vectores representa el ángulo que existe entre estos dos, por lo tanto, se cuantifica la distancia y se pueden identificar vectores similares y vectores distantes. Asimismo, la función *softmax* se encarga de maximizar o minimizar la distancia obtenida entre vectores. Por otro lado, dividir el resultado entre la raíz de $\mathbf{d_k}$ se debe a un proceso de normalización. Finalmente, los resultados se multiplican con $\mathbf{V}$, destacando así las relaciones más relevantes, en consecuencia, los vectores embebidos son enriquecidos con las correspondencias no lineales al ser proyectadas a la matriz de atención, esto resulta en un proceso que evita la alta dependencia local (limitación de las secuencias recurrentes), logrando incluir coherencias del lenguaje que tienen dependencias temporales marcadas, pero que coexisten de forma natural entre las secuencias.

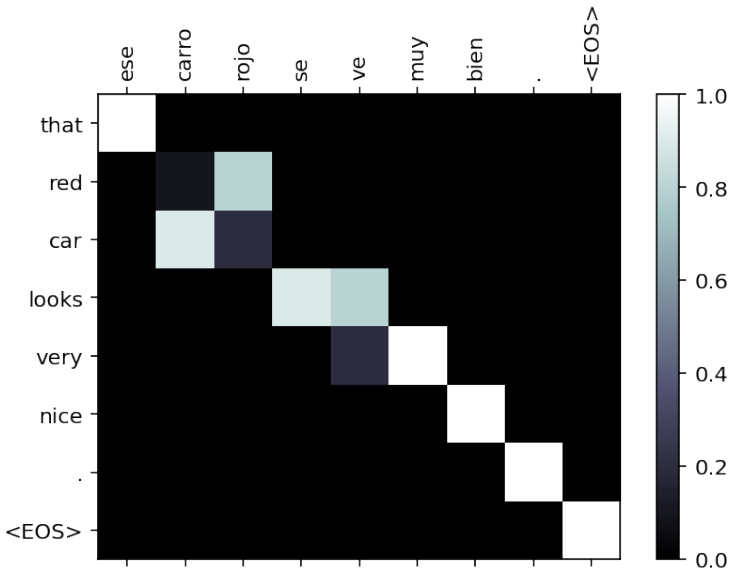
1.2.1.2. Esquema de múltiples atenciones: *multi-head*. La atención de tipo *scale-dot product* obtiene relaciones no lineales entre secuencias, lo cual, representa una pieza clave en el proceso codificación-decodificación. Sin embargo, con base en un conjunto de entrenamiento, una única matriz de atención no puede abarcar las múltiples relaciones que pueden existir en un lenguaje. Debido a esto, en la literatura se han propuesto módulos de procesamiento que involucran múltiples módulos de atención, los cuales son calculados en forma paralela y durante el entrenamiento se dedican a explotar diversas relaciones entre secuencias. El grupo de estos módulos de atención, calculados en un mismo nivel de procesamiento, se denominan mecanismo de atención de múltiples cabezas (*multi-head attention mechanisms*). Para ello, los vectores embebidos de dimensión $\mathbf{d_e}$ se dividen entre un número $\mathbf{h}$ de cabezas, donde $\frac{\mathbf{d_e}}{\mathbf{h}} \in N^+$. Esto, genera un número $\mathbf{h}$ de vectores $\mathbf{K}$, $\mathbf{Q}$ y $\mathbf{V}$, asimismo, se define un número $\mathbf{h}$ de módulos *scale-dot product*, donde cada una de esta terna de vectores se introduce a un módulo de atención.

Dividir entre secciones menores los vectores embebidos permite explotar de manera más local las relaciones temporales entre secuencias, obteniendo así diferentes matrices de atención a través de la dimensión embebida. La figura 3 ilustra un ejemplo de matriz de atención.

1.3. LA LENGUA DE SEÑAS Y LOS SOPORTES PARA LA TRADUCCIÓN AUTOMÁTICA

Múltiples iniciativas computacionales se han dedicado a soportar la interpretación de la LS y su traducción a secuencias de texto. Inicialmente, la metodología de traducción se enfocaba en descriptores para capturar primitivas de imágenes simples y modelar gestos aislados, sin embargo, el modelado del lenguaje se realizaba so-

Figura 3. Ejemplo de matriz de atención entre dos secuencias en un problema de traducción automática neuronal. En el eje x y el eje y se encuentran las palabras de una oración en español y la traducción generada en inglés, respectivamente. Entre el campo más se acerque a 1, más correlación existe entre las dos secuencias.



bre escenarios controlados [10]. Estas restricciones se asocian, entre otras cosas, a los retos inherentes, como la variabilidad textural, las variaciones en los dispositivos ópticos de captura y también al escaso modelado dinámico de los cambios gestuales durante una traducción [11]. Para superar el modelado temporal, se han propuesto algunos enfoques estadísticos para aproximar el reconocimiento continuo de la LS mediante el seguimiento y el modelado de la mano dominante del intérprete [12]. Estos

---

[10] Helen Cooper y col. “Sign language recognition using sub-units”. En: *The Journal of Machine Learning Research* 13.1 (2012), págs. 2205-2231.

[11] Lionel Pigou y col. “Sign language recognition using convolutional neural networks”. En: *European Conference on Computer Vision*. Springer. 2014, págs. 572-578.

[12] Oscar Koller, Jens Forster y Hermann Ney. “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers”. En: *Computer Vision and Image Understanding* 141 (2015), págs. 108-125.

modelos se han diseñado generalmente bajo la hipótesis de un modelo de Markov oculto, consiguiendo incluir algunas consideraciones gramaticales y relaciones locales en las frases [10]. Sin embargo, estas estrategias tienen limitaciones de flexibilidad para incluir las múltiples variantes del lenguaje, y a su vez, generalmente su carácter temporal es limitado.

En la actualidad, las representaciones más eficaces para la traducción automática de la LS se basan en arquitecturas de aprendizaje profundo, donde se integran complejas estrategias de codificación-decodificación de la información visual y textual. Por ejemplo, Cui et al. [13] propusieron una red neuronal recurrente y convolucional para integrar la información visual (representación convolucional) y explotar la información temporal (estrategia recurrente). Este enfoque híbrido proporciona una aproximación para capturar las dependencias temporales, pero sólo en intervalos cortos, con un nivel de reconocimiento de glosas insuficiente en algunas muestras. Por ello, Camgoz et al. [3] introdujeron el problema de la traducción de la LS, contextualizado a partir de una arquitectura codificadora-decodificadora que incluye módulos recurrentes, convolucionales y de atención, los cuales, de manera conjunta, recuperan las dependencias no lineales durante la traducción. Este trabajo muestra fuertes ventajas para incluir dependencias lingüísticas relativamente largas, sin embargo, su representación sigue basándose en secuencias de fotogramas crudas, lo que hace que el proceso de entrenamiento y optimización sea significativamente costoso.

Otros enfoques han abordado esta limitación a través del flujo óptico y la pose cor-

---

[13] Runpeng Cui, Hu Liu y Changshui Zhang. “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization”. En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, págs. 7361-7369.

poral [14][15], debido a que estas representaciones apoyan la codificación del lenguaje, ya que se basan en una representación gestual específica e incluyen un marcador muy importante del lenguaje, por ejemplo, la información cinemática. Esta fuente proporciona por sí misma información de referencia en la traducción, pero también resulta más compacta y fiable en los esquemas de entrenamiento. No obstante, estas arquitecturas pierden representaciones intermedias durante la traducción de vídeo a texto, lo que da lugar a algunas traducciones de texto incoherentes y con cierta tendencia a sobreaprender las proyecciones entre ambos dominios. Cheng et al. [16] proponen una arquitectura compuesta únicamente por redes convolucionales que incluyen una representación intermedia y evitan el aprendizaje exhaustivo de una configuración recurrente. Este enfoque aprovecha las representaciones convolucionales en una dimensión, pero asume el intervalo temporal de cada glosa como un promedio de ventanas, dejando de lado la alta variabilidad de dicho intervalo temporal durante una comunicación real. Específicamente, durante el desarrollo de cada una de las oraciones en LS pueden existir diferentes duraciones de acuerdo a las interpretaciones del autor, la frase realizada y las variaciones culturales del lenguaje.

Recientemente, los *transformer* han sido arquitecturas ideales para tratar las dependencias a largo plazo de las estrategias de traducción. Estas metodologías se han adaptado en la traducción de LS, considerando una representación profunda que sólo tiene en cuenta los módulos de atención para convertir vídeos a secuen-

---

[14] Jefferson Rodriguez y Fabio Martínez. “How important is motion in sign language translation?” En: *IET Computer Vision* 15.3 (2021), págs. 224-234.

[15] Mathieu De Coster, Mieke Van Herreweghe y Joni Dambre. “Sign language recognition with transformer networks”. En: *12th International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA). 2020, págs. 6018-6024.

[16] Ka Leong Cheng y col. “Fully convolutional networks for continuous sign language recognition”. En: *European Conference on Computer Vision*. Springer. 2020, págs. 697-714.

cias de texto [5][17][6][18]. Estas estrategias han demostrado una notable capacidad para capturar dependencias no lineales y construir espacios embebidos, donde la información de video y de texto convergen para encontrar relaciones. Estos enfoques han superado al estado del arte, no obstante, siguen mostrando limitaciones a la hora de enfrentarse a un escenario real, donde la inclusión de nuevas expresiones aumenta exponencialmente la complejidad del problema, debido a la alta riqueza en el vocabulario de la LS, junto con las múltiples versiones de gestos para expresar las mismas ideas. Por esta razón, es desafiante agrupar dichos signos en glosas coherentes que puedan reducir la complejidad computacional de la representación y puedan ser desplegadas en escenarios reales. En este sentido, Yin y Read [6] incluyen las glosas realizando la traducción en dos pasos, obteniendo la traducción final con base en las glosas. Esto representa un cuello de botella de información entre los dos pasos, ya que la traducción final está condicionada por la calidad del reconocimiento de glosas. Camgoz et al. [5] en su investigación incluyen las glosas como una representación intermedia, sin embargo, las características visuales que se utilizan son vectores embebidos pre-entrenados basados en videos RGB, lo que indica que no se está incorporando una representación visual que pueda ayudar a mejorar el modelamiento de esta lengua.

---

[17] Ben Saunders, Necati Cihan Camgoz y Richard Bowden. “Progressive transformers for end-to-end sign language production”. En: *European Conference on Computer Vision*. Springer. 2020, págs. 687-705.

[18] Kayo Yin y Jesse Read. “Attention is all you sign: sign language translation with transformers”. En: *Proceedings of the European Conference on Computer Vision (ECCV) Workshop on Sign Language Recognition, Translation and Production (SLRTP)*. Vol. 23. 2020.

2. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

La pérdida auditiva y los problemas en general de comunicación implican dramáticas limitaciones de interacción en la sociedad, lo cual impacta de forma significativa en la vida cotidiana. La comunicación mediante la LS es uno de los canales más efectivos que han adoptado las personas con discapacidades auditivas. Sin embargo, esta lengua no es de uso común por el resto de la sociedad, lo que conlleva a que siga existiendo una clara limitación comunicativa que afecta a la comunidad sorda. Incluso, diferentes factores culturales y diversidades en el lenguaje hacen que este mecanismo de comunicación no tenga poblaciones demográficamente significativas, lo que conlleva a diversidad e independencia de la lengua entre regiones cercanas, dificultando incluso la comunicación entre personas sordas. Es por ello, que tecnologías emergentes pueden ayudar a generar mecanismos de traducción automática, que garanticen la comunicación de personas sordas con el resto de la sociedad.

El reconocimiento y traducción de la LS es un problema que pretende sincronizar representaciones visuales de video con secuencias de texto en el lenguaje escrito. Para ello, en la actualidad se cuentan con diferentes métodos computacionales que han aproximado este problema en corpus restringidos a una LS particular. Sin embargo, existen diversos desafíos inherentes de la lengua, como múltiples variantes en diferentes países, variaciones en la representación gestual y un enriquecido y redundante vocabulario. Asimismo, estas representaciones tienen que lidiar con variantes en la proyección gesto-visoespacial registrada en los videos, dado que, al haber sido realizadas por diferentes intérpretes, existe una alta variabilidad en los tiempos de la mímica y en los movimientos de los gestos.

Sumado a lo anterior, la naturaleza ágrafa en la LS limita las estrategias computacionales por la alta asincronía entre los dos modos de información (gestual y escrita).

De hecho, mientras la información gestual es densa, redundante y multiarticular, la información que corresponde a nivel textual es escasa y con fuertes variantes gramaticales. Por lo tanto, un desafío principal hoy en día es hacer correspondencias y proyecciones entre secuencias con naturalezas y reglas gramaticales diferentes. Entonces, una alternativa podría ser el uso de glosas como interpretaciones intermedias de la LS. Estas glosas agrupan espaciotemporalmente gestos de acuerdo con un conjunto de símbolos textuales, donde las representaciones espaciotemporales obtenidas pueden ser clave para aliviar costos computacionales, evitar sesgos en las representaciones gestuales y requerimientos de grandes volúmenes de datos para modelar la alta variabilidad del lenguaje.

3. OBJETIVOS

Objetivo general

- Desarrollar una estrategia de aprendizaje profundo que incluya una representación intermedia de glosas para soportar la traducción de la lengua de señas de video a texto.

Objetivos específicos

1. Seleccionar un conjunto de datos de la lengua de señas que contenga secuencias de video y anotaciones relacionadas con las traducciones en glosa y traducciones correspondientes.
2. Implementar un módulo de atención a nivel de glosa, el cual se emplee como una representación intermedia que facilite la transición entre el video y el texto.
3. Integrar el modelo de representación intermedia de glosas en un modelo profundo para la traducción de la lengua de señas.
4. Evaluar la capacidad de la representación intermedia basada en glosas para realizar la traducción de video a texto en el contexto de la lengua de señas.

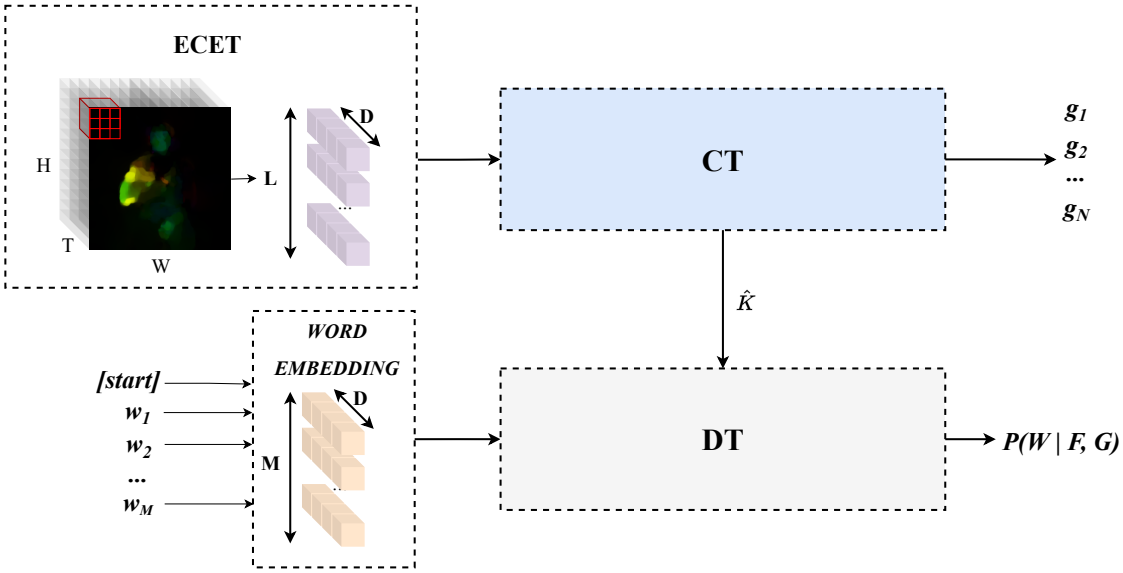
4. MÉTODO PROPUESTO

Esta sección introduce la estrategia propuesta, basada en una representación *transformer* multitarea *end-to-end* que permite la traducción de LS y el reconocimiento de glosas a partir del flujo óptico de videos de LS. Para ello, cada secuencias de vídeo $\mathbf{V} = (v_1, \cdots, v_T)$ con $\mathbf{V} \in \mathbb{N}^{T \times W \times H \times C}$, donde T es la longitud temporal, $(W \times H)$ es la resolución espacial del fotograma y C hace referencia al número de canales de cada fotograma, corresponde a una frase en LS que puede describirse con N glosas como $\mathbf{G} = (g_1, \cdots, g_N)$. Asimismo, cada frase a nivel de glosa tiene una proyección con una frase del lenguaje escrito $\mathbf{W} = (w_1, \cdots, w_M)$ con M palabras. Esto indica que las expresiones en LS tienen correspondencias en tres diferentes representaciones: $\mathbf{V} \rightarrow \mathbf{G} \rightarrow \mathbf{W}$. Este modelo tiene la capacidad de estimar la correspondencia del lenguaje hablado, es decir, es capaz de determinar la siguiente palabra de una frase dada una representación de glosa intermedia y la información de vídeo; esto se puede definir como $P(W_t|V, G, W_{t-1})$.

Para ello, el enfoque propuesto recibe como entrada una secuencia de videos de LS en una representación de flujo óptico (sección 4.1). Luego, la información de video se codifica en vectores embebidos cinemáticos a través de una representación convolucional volumétrica profunda (sección 4.2). A continuación, estos embebidos con referencias de localización temporal funcionan como entrada a un módulo de atención *multi-head*, el cual, calcula una representación profunda que recupera las principales relaciones no lineales de la información cinemática (sección 4.3). Esta representación profunda permite obtener un reconocimiento de glosas escritas $\mathbf{G}$, y además, un conjunto de embebidos que se introducen a un módulo de atención que los relaciona con el lenguaje escrito. De forma complementaria, un decodificador *transformer* consigue una representación profunda de atención entre las secuencias de texto y los embebidos que contienen información de glosas y movimiento (sección

4.4). Este último módulo genera la traducción con base en las frases en LS y con el apoyo de la representación intermedia basada en glosas. El esquema general del método propuesto se muestra en la figura 4.

Figura 4. Esquema del método propuesto. El flujo óptico se da como entrada a una estrategia volumétrica (ECET) que extrae a bajo nivel la información cinemática del video. Esta información se procesa a través del codificador (CT), el cual, a través de relaciones no lineales entre la información espaciotemporal, modela y reconoce las glosas. Por otro lado, el decodificador (DT) genera la traducción basándose en proyecciones temporales entre los videos y el lenguaje escrito. **ECET:** Extractor de características espaciotemporales, **CT:** Codificador *transformer*, **DT:** Decodificador *transformer*.



4.1. DEL VIDEO A UNA REPRESENTACIÓN DE FLUJO ÓPTICO

La LS se basa en componentes gestuales y visuales, donde no sólo la configuración geométrica define la comunicación, sino que también el movimiento incluye un

significado no visual que hace parte del lenguaje [19]. En este sentido, es recomendable primero calcular una representación espaciotemporal que considere los cambios de forma, y a su vez, incluya primitivas cinemáticas durante la representación de la LS. Además, la transformación de una secuencia de video $\mathbf{V} = (v_1, \cdots, v_T)$ en una representación cinemática $\mathbf{F} = (f_1, \cdots, f_T)$ puede evitar la alta variabilidad espacial de los intérpretes de la LS, debido a que esta representación resulta en una alternativa visual menos densa, ya que elimina información de fondo y resalta características que hacen referencia al movimiento. Por estas razones, el enfoque propuesto comienza calculando una representación de movimiento aparente de $\mathbf{V}$. La transformación de videos a una representación densa de movimiento $(\mathbf{V} \rightarrow \mathbf{F})$ se consigue calculando un flujo óptico denso a lo largo de cada una de las secuencias. Particularmente, en este trabajo se implementó el flujo óptico de Brox [20], el cual tiene en cuenta los grandes desplazamientos, una de las características más relevantes para la LS. Para ello, para cada par de fotogramas consecutivos (v_t, v_{t+1}) se calcula un campo de velocidad (ν_t) que resulta de un proceso típico de minimización, considerando el error aparente ($E_a(\nu) = |v_t - v_{t+1}|^2$), el error estructural de gradiente ($E_g(\nu) = |\nabla v_t - \nabla v_{t+1}|^2$) y el campo de flujo estructural parcial $E_s(\nu) = |\nabla u|^2$. Además, los grandes desplazamientos se calculan a partir de una implementación no local, la cual se basa en patrones similares de campos de velocidad entre fotogramas, determinando puntos claves entre los elementos de la secuencia. A partir de estas operaciones se obtiene una representación de la secuencia volumétrica que corresponde con campos densos consecutivos $\mathbf{F} = (\nu_1, \cdots, \nu_{T-1}) \in \mathbb{N}^{(T-1)\times W^{'}\times H^{'}\times C^{'}}$,

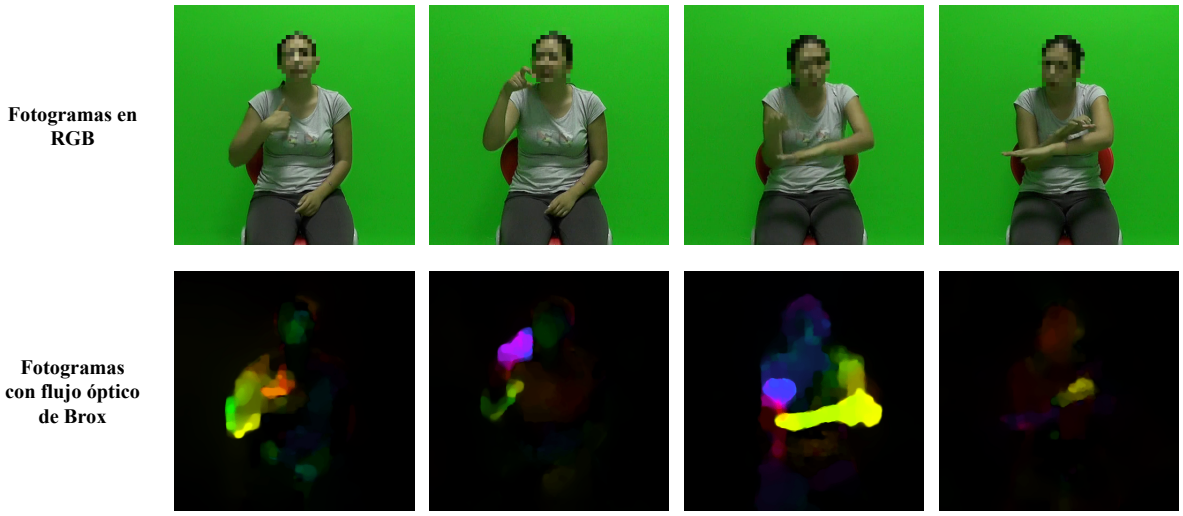
---

[19] Wendy Sandler. “The phonological organization of sign languages”. En: *Language and linguistics compass* 6.3 (2012), págs. 162-182.

[20] Thomas Brox y Jitendra Malik. “Large displacement optical flow: descriptor matching in variational motion estimation”. En: *IEEE transactions on pattern analysis and machine intelligence* 33.3 (2010), págs. 500-513.

donde $(T-1)$ corresponde al total de campos densos calculados para cada secuencia, $(W' \times H')$ son la dimensión espacial para cada campo denso recuperado y C' es el número de canales de cada fotograma. En la figura 5 se muestra un ejemplo de un conjunto de muestras de $\mathbf{V}$ en una representación $\mathbf{F}$.

Figura 5. Ejemplo de representación del flujo óptico de Brox. Un conjunto de fotogramas de un video de LS representado en el flujo óptico de Brox. Cada fotograma en esta representación contiene características visuales que resaltan el movimiento.



4.2. REPRESENTACIÓN VOLUMÉTRICA ESPACIO-TEMPORAL (ECET)

Una vez obtenida la representación cinemática $\mathbf{F}$ de la LS, en cada secuencia de video se hace una proyección a vectores embebidos que codifiquen la información del gesto $(\mathbf{F} \rightarrow \mathbf{K})$, en vectores de baja dimensionalidad k_t en cada instante de tiempo. La proyección de la representación $\mathbf{F} \in \mathbb{N}^{(T-1)\times W' \times H' \times C'}$ en $\mathbf{K} \in \mathbb{N}^{L \times d}$ se logra a través de una estrategia volumétrica convolucional, que procesa volúmenes y progresivamente reduce la dimensionalidad hasta descriptores unidimensionales, donde cada vector embebido $L \in \mathbb{R}^d$. Inspirados en la arquitectura LTC (*long-term*

temporal convolutions [21]), el modelo aprende transformaciones no lineales que, de forma progresiva, computan una representación compleja expresada en proyecciones de alto nivel. Por lo tanto, el uso de esta estrategia garantiza la cuantificación de información cinemática de los videos, dado que, al ser una secuencia de convoluciones altamente correlacionadas, el último bloque volumétrico representa un espacio embebido $\mathbf{K}$ que modela de forma robusta la información espaciotemporal de la lengua. La figura 6 muestra la estrategia volumétrica implementada, la cual entrega un conjunto de vectores embebidos ocultos que cuantifican las relaciones complejas presentes en la representación cinemática.

Como se observa en la figura 7, la arquitectura convolucional volumetrica está compuesta por seis bloques convolucionales 3D, donde cada bloque está conformado por una secuencia de capas definidas, cuyo orden es *3DCNN* $\rightarrow$ *batch normalization* $\rightarrow$ *reLU* $\rightarrow$ *max pooling 3D*. Para ello, la capa 3DCNN obtiene representaciones densas sobre la secuencia $\mathbf{F}$ a partir de cálculos convolucionales. El *batch normalization* regulariza la información embebida, conllevando a un mejor proceso de aprendizaje y reduciendo de forma significativa el tiempo de entrenamiento. Por otro lado, la activación *reLU* aplica una función no lineal sobre los datos, permitiendo a la red determinar patrones no-lineales, con sesgo en las representaciones positivas. Asimismo, la operación *max pooling 3D* reduce la dimensionalidad del tensor, y a su vez, preserva la información relevante. En este caso, la dimensión temporal de los vídeos de entrada T tiene una longitud diferente con respecto a L.

---

[21] Gül Varol, Ivan Laptev y Cordelia Schmid. “Long-term temporal convolutions for action recognition”. En: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), págs. 1510-1517.

Figura 6. Extractor de características espaciotemporales (ECET). Este módulo propuesto extrae características cinemáticas de bajo nivel y a largo plazo utilizando una estrategia volumétrica. Cada secuencia resulta en un espacio latente con dimensiones $L \times d$ que contiene la información más relevante del video.

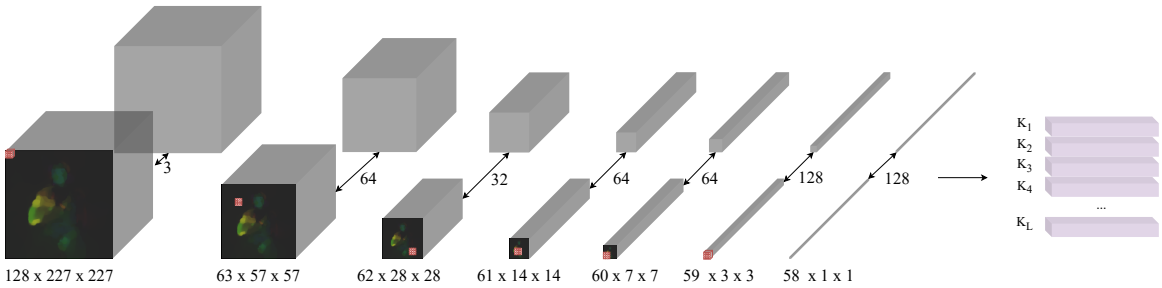
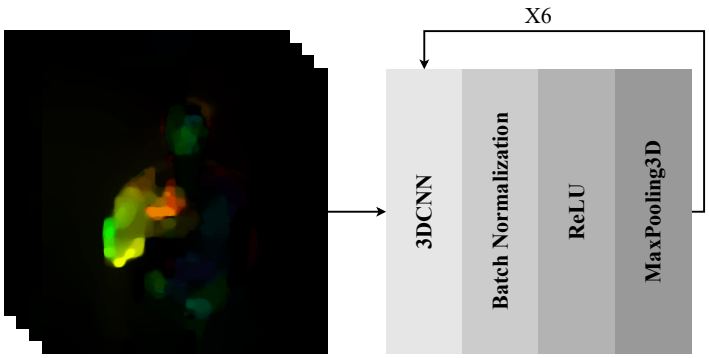


Figura 7. Bloque convolucional del ECET. Cada bloque está compuesto por cuatro capas que cumplen diferentes funciones. En la metodología propuesta se emplearon seis bloques consecutivos.

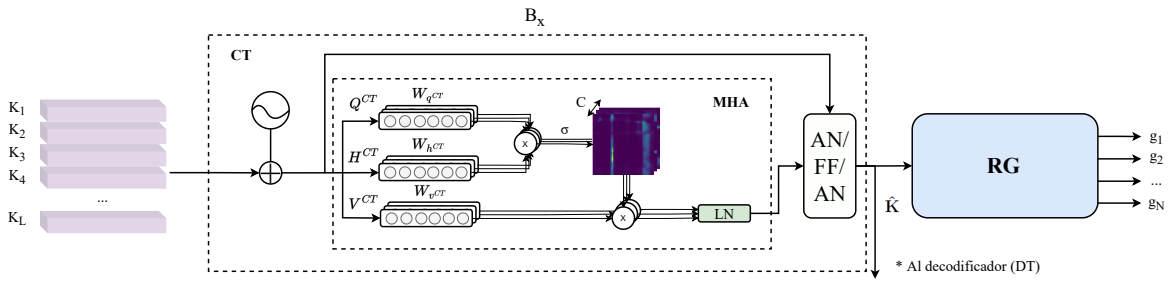


4.3. CODIFICADOR *TRANSFORMER* (CT)

El problema principal de este módulo es establecer relaciones entre información cinemática embebida en una representación de baja dimensionalidad y glosas escritas ($\mathbf{K} \rightarrow \mathbf{G}$). Para ello, se implementó un codificador *transformer* (CT) que tiene como objetivo modelar y reconocer un conjunto de glosas dada una representación cinemática embebida del video de entrada, expresado como $P(g_1, \cdots, g_N | k_1, \cdots k_L)$. Por lo tanto, este módulo actúa como un codificador de información espaciotemporal de $\mathbf{K}$ y de representaciones intermedias de glosas $\mathbf{G}$ en la LS. La figura 8, ilustra

los principales componentes utilizados en este módulo, donde las capas con múltiples autoatenciones enriquecen la representación temporal de los embebidos y se proyectan a través de una representación neuronal no lineal.

Figura 8. Codificador *transformer*. Este módulo determina un conjunto de embebidos $\hat{\mathbf{K}}$ de información codificada y un conjunto $\mathbf{G}$ de glosas. Para ello, con el espacio embebido $\mathbf{K}$ como entrada, una capa *positional encoding* inserta información posicional a cada vector K_l, luego, esta representación pasa a través de un módulo de atención *multi-head*, el cual encuentra relaciones no lineales entre cada elemento de $\mathbf{K}$. Finalmente, la información se da como entrada a una capa *point-wise feed Forward* seguida de módulos de normalización. Esta secuencia se repite B veces, obteniendo así una matriz $\hat{\mathbf{K}}$ con información cinemática codificada. **MHA:** Atención *multi-head*, **AN:** Adición y normalización, **FF:** Capa *point-wise feed-forward*, **LN:** Capa lineal, σ : Activación *softmax*, **RG:** Módulo de reconocimiento de glosas.



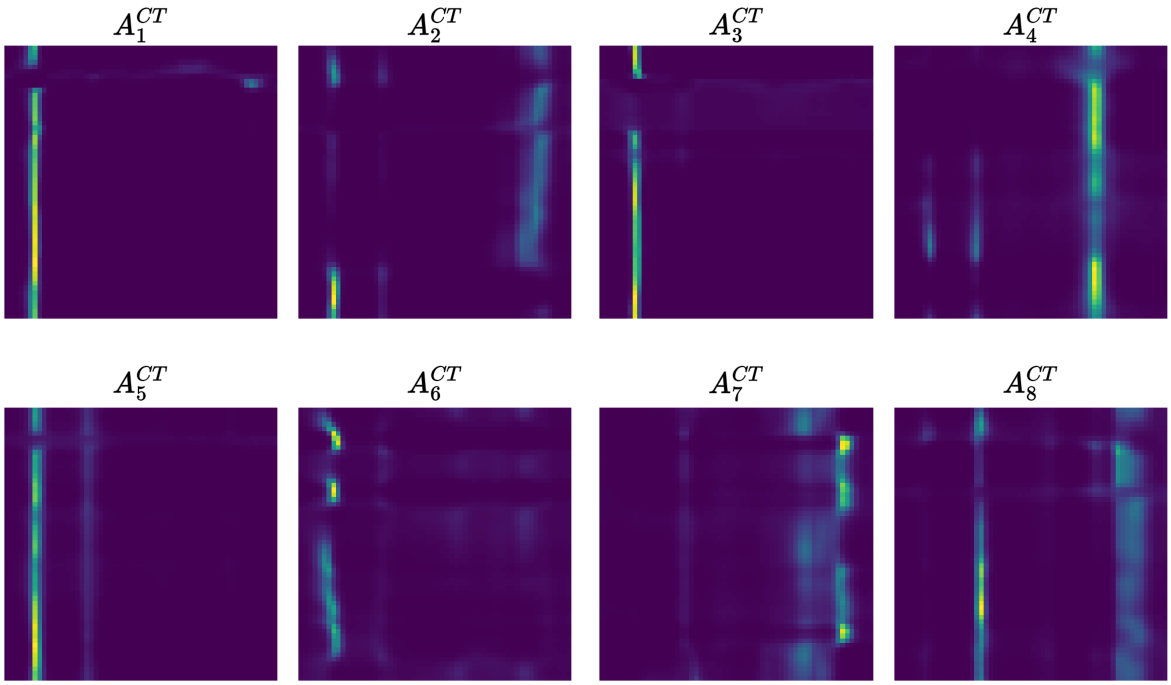
4.3.1. Codificación posicional absoluta. El primer paso consiste en agregar información posicional a cada vector embebido de $\mathbf{K}$. Esto es necesario debido a que la estrategia *transformer* no emplea metodologías recurrentes, lo que indica que los datos de entrada ingresan de forma paralela. Por lo tanto, al ingresar en un solo paso de tiempo no es posible para la red determinar información posicional de cada embebido. En este trabajo se utilizó la codificación posicional absoluta [4], la cual utiliza un algoritmo que emplea funciones sinusoidales que garantizan un valor único para cada vector embebido. Por lo tanto, el proceso de agregar información temporal a cada embebido de $\mathbf{K}$ se define como: $\mathbf{K} = \mathbf{K} + PositionalEncoding(\mathbf{K})$.

4.3.2. Mecanismos de múltiple atención (MHA) (*multihead attention*) en un dominio de glosas cinemáticas. La representación de glosas desde vectores embebidos es lograda por la proyección del conjunto de embebidos de $\mathbf{K}$ en mecanismos múltiples de atención (MHA, *multihead attention*, por sus nombre en inglés). Específicamente, los módulos MHA determinan relaciones no lineales entre dos dominios, con el fin de obtener múltiples atenciones que recuperen relaciones no locales entre las dos secuencias. En este caso, la metodología *transformer* propone aplicar estos módulos de atención sobre un solo dominio, buscando correlacionar cada elemento K_l con los demás de la secuencia. Este tipo de atención se denomina autoatención (comúnmente conocido como: *self-attention*). Entonces, un bloque de múltiple atención corresponde a la implementación en paralelo de diversos mecanismos de autoatención en un mismo nivel de procesamiento.

Particularmente, en este trabajo se implementó un bloque de múltiples atenciones con C módulos de autoatención. Para cada módulo de atención $\mathbf{O}_i^{CT}$, se realiza una proyección del espacio embebido $\mathbf{K}$ en tres diferentes componentes de información, con el propósito de modelar glosas a partir de la información cinemática obtenida. Primero, se hacen las proyecciones típicamente conocidas como: *key* ($\mathbf{H}_i^{CT} = W_{h^{CT}}\mathbf{K}$) y *query* ($\mathbf{Q}_i^{CT} = W_{q^{CT}}\mathbf{K}$) proyectando entradas cinemáticas $\mathbf{K}$ a través de los pesos $W_{h^{CT}}$ y $W_{q^{CT}}$, respectivamente, donde $W_{h^{CT}}, W_{q^{CT}}, \in \mathbb{R}^{L\times\frac{d}{C}}$. Con estas proyecciones es posible calcular mapas de atención $\mathbf{A}_i^{CT} = \sigma((\mathbf{H}_i^{CT})^T\mathbf{Q}_i^{CT})$ que codifican correlaciones no lineales entre los vectores embebidos, donde σ es la función de activación *softmax*. Es decir, cada uno de los mapas de atención $\mathbf{A}_i^{CT}$ tiene una representación no lineal que define las principales relaciones entre dos secuencias, donde se rescatan dependencias temporales sin importar la distancia temporal entre los elementos. La figura 9 ilustra un ejemplo de las matrices de atención $\mathbf{A}_i^{CT}$ obtenidas en este módulo.

Por otra parte, en paralelo con la proyecciones $(\mathbf{H}^{CT}, \mathbf{Q}^{CT})$ se hizo una proyección

Figura 9. Mapas de atención de glosas $\mathbf{A}^{CT}$ con base en la información cinemática $\mathbf{K}$. Por cada cabeza se determina un mapa con diferentes relaciones entre la secuencia. Este ejemplo cuenta con ocho cabezas y dos capas de codificación. La información extraída corresponde a la segunda capa.



value: ($\mathbf{V}_i^{CT} = W_{v^{CT}}\mathbf{K}; W_{v^{CT}} \in \mathbb{R}^{L\times\frac{d}{C}}$) que es ponderada con respecto a la matriz de atención $\mathbf{A}_i^{CT}$. Con esta ponderación, se obtiene una nueva representación de vectores embebidos, resultando en un mapa de características $\mathbf{O}_i^{CT}$ que destaca los principales componentes locales y no lineales con mayor correspondencia con las glosas. Esta representación se calcula como $\mathbf{O}_i^{CT} = \mathbf{A}_i^{CT}\mathbf{V}_i^{CT}$; $\mathbf{O}_i^{CT} \in \mathbb{N}^{L\times\frac{d}{C}}$. Para un bloque de C mecanismos de atención, se realiza este proceso en múltiples mecanismos entrenados de forma independiente, recuperando por lo tanto C representaciones cinemáticas de atención. $\{\mathbf{O}_1^{CT}, \mathbf{O}_2^{CT} \ldots, \mathbf{O}_C^{CT}\}$. Durante el entrenamiento, la representación múltiple enriquece la representación de las secuencias, logrando que cada mecanismo de atención se enfoque en relaciones específicas no-lineales que contribuyen a la apropiada correspondencia entre vecto-

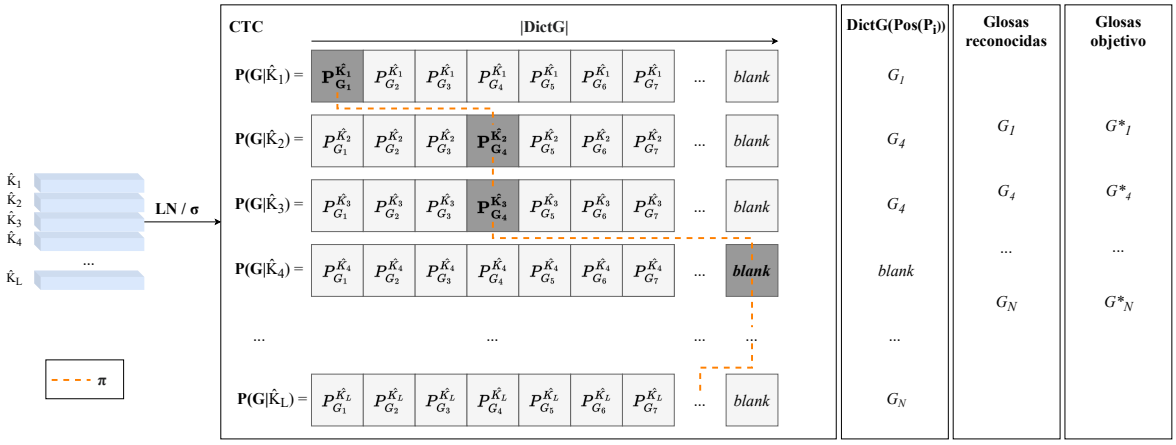
res cinemáticos, desde las secuencias de video y las glosas. Estas representaciones cinemáticas se concatenan y se proyectan a un vector de representación (MH^{CT} = concat($\mathbf{O}_1^{CT}, \mathbf{O}_2^{CT} \ldots, \mathbf{O}_C^{CT}$)$W_{Mh^{CT}}$), según un ajuste de pesos $W_{Mh^{CT}}$. Estas representaciones se dan como entrada a través de dos capas densas lineales, obteniendo una representación final $\hat{\mathbf{K}} \in \mathbb{R}^{L \times d}$, donde $\hat{\mathbf{K}}$ representa los vectores embebidos mejorados a través de la proyección de los módulos MHA. Este procesamiento representa una capa B_x^{CT} de codificación de tipo *transformer* en este trabajo. Esta capa puede ser replicada secuencialmente para lograr un mayor procesamiento y representación de los datos.

4.3.3. Módulo de reconocimiento de glosas (RG). Una vez obtenida una representación enriquecida $\hat{\mathbf{K}}$, por los módulos de atención, en este trabajo se realizó una correspondencia con una representación intermedia, es decir, con un diccionario de glosas $\mathbf{G}$, buscando una alineación coherente y más simple entre los gestos representados en video y la correspondencia escrita. En este sentido, una unidad de glosa escrita $g_n = \{V_i\}_{i:i+T'} \sim \{\hat{K}_i\}_{i:i+L'}$ describe un texto como un subconjunto de fotogramas ΔT, donde se realiza el gesto, lo que se puede aproximar por un subconjunto de embebidos ΔL en nuestra representación $\hat{\mathbf{K}}$.

Entonces, para lograr esta correspondencia ($\hat{\mathbf{K}} \rightarrow \mathbf{G}$), en este trabajo se implementó un módulo de CTC (en inglés, *connectionist temporal classification*) que permite determinar glosas sin necesidad de tener anotaciones explícitas en la dimensión temporal. El objetivo de este módulo es reconocer un conjunto de glosas objetivo $\mathbf{G}^*$, dada la representación embebida $\hat{\mathbf{K}}$, definida como: $P(\mathbf{G}^*|\hat{\mathbf{K}})$. En este caso, esta relación fue implementada como el camino más probable entre vectores $\hat{K}_i$ y el diccionario de glosas, como: $P(\mathbf{G}^*|\hat{\mathbf{K}}) = \sum_{\pi \in \beta} P(\pi|\hat{\mathbf{K}})$, donde π representa un camino que pertenece al conjunto β de caminos que conducen a las glosas objetivo $\mathbf{G}^*$. La figura 10 muestra cómo opera el CTC sobre un conjunto $\hat{\mathbf{K}}$ de embebidos. En este caso, el conjunto de embebidos $\hat{\mathbf{K}}$ se proyectan en una capa lineal + *softmax*,

la cual tiene como fin ajustar la dimensionalidad acorde al tamaño del diccionario de glosas y establecer una probabilidad de correspondencia. Entonces se obtiene por cada embebido, la probabilidad correspondiente con respecto a cada Glosa definida en el diccionario (filas $P(G|\hat{K}_i)$). El camino π más probable, entonces corresponde a las glosas con mayor probabilidad en cada embebido, como se ilustra en la figura 10. Una vez se calcula $P(\mathbf{G}^*|\hat{\mathbf{K}})$, el error del módulo CT se determinar a partir de: $\mathbb{L}_{CT} = 1 - P(\mathbf{G}^*|\hat{\mathbf{K}})$ Obteniendo así el valor de la pérdida para el modelamiento de las glosas. Por otra parte, es importante señalar que el conjunto de vectores $\hat{\mathbf{K}}$ es una representación primaria codificada de glosas con información cinemática, la cual permite relacionar en un módulo MHA el dominio escrito $\mathbf{W}$ con $\mathbf{F}$.

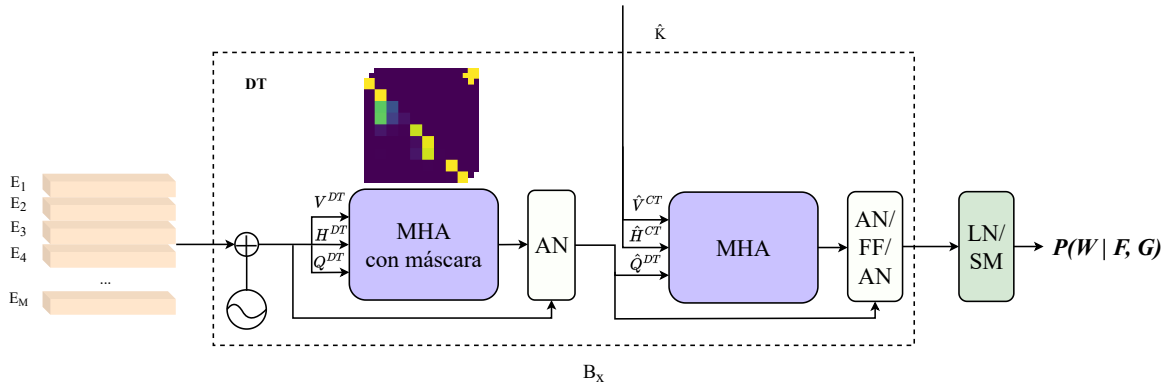
Figura 10. Módulo *Connectionist Temporal Classification (CTC)*. El espacio embebido $\hat{\mathbf{K}}$ pasa a través de una capa lineal (LN) y una activación *softmax* (σ). La secuencia de casillas resaltadas corresponde a un camino $\pi \in \beta$; en algunos pasos l el módulo no reconoce ninguna glosa, por lo tanto, se asocia el elemento especial $blank$. Luego, a partir del diccionario de glosas y del índice de la dimensión embebida seleccionada, se obtiene una glosa escrita por cada paso de tiempo l. Luego, se realiza un proceso de decodificación que busca eliminar glosas repetidas y caracteres marcados como $blank$.



4.4. DECODIFICADOR *TRANSFORMER* (DC)

El codificador propuesto permite la proyección y obtención de representaciones embebidas $\hat{\mathbf{K}}$ y glosas $\mathbf{G}$, dada una secuencia de entrada $\mathbf{V} \rightarrow (\hat{\mathbf{K}}, \mathbf{G})$. A partir de dichas representaciones intermedias, se plantea entonces lograr una proyección a la secuencia textual correspondiente $\mathbf{W}$. En este trabajo, se implementó un módulo decodificador *transformer* (DC), el cual busca modelar el dominio de palabras $\mathbf{W}$ y encontrar proyecciones entre el video y el lenguaje escrito utilizando las representaciones intermedias $(\hat{\mathbf{K}}, \mathbf{G})$. A continuación se detallan los módulos que componen el decodificador *transformer* implementado en este trabajo (una ilustración está disponible en la figura 11).

Figura 11. Decodificador *transformer* (DT). Este módulo tiene como objetivo predecir un conjunto de palabras $\mathbf{W} = (w_2 \cdots w_M)$. Para ello, un espacio latente $\mathbf{E}$, el cual es una representación embebida de las palabras $\mathbf{W}$, se da como entrada a un decodificador *transformer* (DT), el cual explota relaciones temporales entre los elementos de $\mathbf{W}$ y determina proyecciones entre $\mathbf{F}$ y $\mathbf{W}$ a través de módulos MHA y la información codificada en $\hat{\mathbf{K}}$. Esta secuencia se repite B veces, donde finalmente el espacio latente calculado pasa a través de una capa lineal y una capa *softmax*, dando como salida $P(w_m|F, G, w_{m-1})$ en cada paso de decodificación.



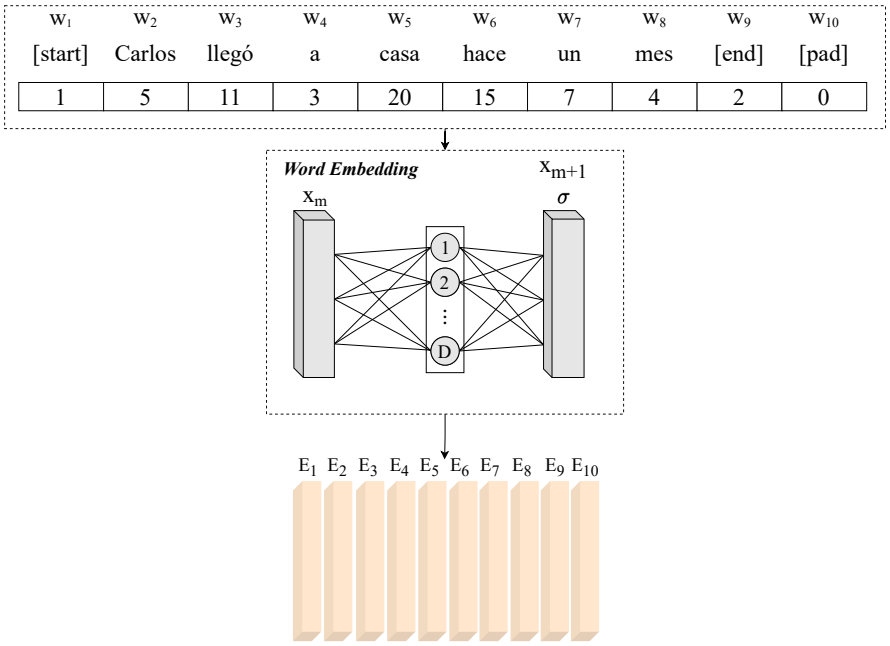
4.4.1. Capa *word embedding*. En este módulo decodificador se obtiene un espacio latente que representa el conjunto de palabras $\mathbf{W}$. Este espacio latente permi-

te obtener descriptores de palabras que codifican la información semántica a partir de proyecciones embebidas, es decir, las palabras quedan posicionadas según su significado. En este caso, a cada frase $\mathbf{W}$ se le agregan las palabras especiales (*tokens*) $[start]$ y $[end]$ al inicio y final de cada oración. De esta manera se puede incluir información implícita sobre la longitud de las secuencias y las palabras más ocurrentes en los inicios y finales de cada frase. Para ello, se realiza un proceso de *padding* en el conjunto de oraciones $\mathbf{W}$, buscando que todas las secuencias escritas tengan la misma longitud, así como su respectiva representación numérica por cada elemento w_m. Entonces, la representación embebida $\mathbf{E} = WordEmbedding(W)$ se implementa como una red neuronal densa simple, con una única capa intermedia (soporte de los vectores embebidos) y una capa de salida que aprende la probabilidad de ocurrencia de palabras vecinas. En este caso, la proyección de cada frase, resulta en una codificación embebida $\mathbf{E} \in \mathbb{R}^{M \times D}$ que corresponde a la representación de $\mathbf{W}$, como se ilustra en la figura 12.

A su vez, a cada elemento de $\mathbf{E}$ se le agrega información posicional, utilizando la estrategia *positional encoding*, como: $\mathbf{E} = \mathbf{E} + PositionalEncoding(\mathbf{E})$.

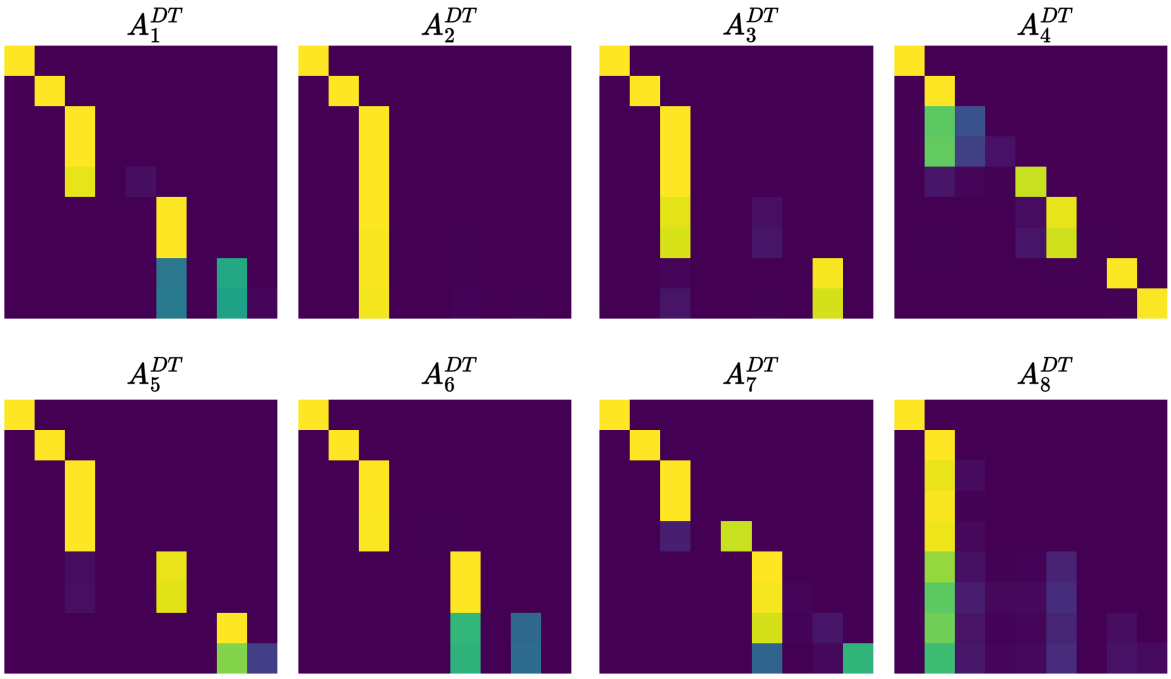
4.4.2. Módulos MHA entre LS y lenguaje escrito. El modelamiento del lenguaje escrito se determina a partir de proyecciones encontradas en módulos MHA, donde en este caso, el objetivo del módulo es determinar un conjuntos de características $\mathbf{O}_i^{DT}$ que recuperan relaciones temporales no lineales entre los elementos del conjunto $\mathbf{E}$ de embebidos. Para ello, las proyecciones *key* ($\mathbf{H}_i^{DT} = W_{h^{DT}}\mathbf{E}$), *query* ($\mathbf{Q}_i^{DT} = W_{q^{DT}}\mathbf{E}$) y *value* ($\mathbf{V}_i^{DT} = W_{v^{DT}}\mathbf{E}$) se introducen a un módulo de múltiples autoatenciones, donde los pesos $W_{q^{DT}}, W_{h^{DT}}, W_{v^{DT}} \in \mathbb{R}^{M \times \frac{d}{C}}$. Asimismo, los mapas de atención $\mathbf{A}_i^{DT} = \sigma((\mathbf{H}_i^{DT})^T\mathbf{Q}_i^{DT})$ cuantifican desde diferentes perspectivas las relaciones temporales que existen entre el conjunto de palabras, representada como embebidos. Posteriormente se determina el conjunto de características $\mathbf{O}_i^{DT} = \mathbf{A}_i^{DT}\mathbf{V}_i^{DT}$; $\mathbf{O}_i^{DT} \in \mathbb{N}^{M \times \frac{d}{C}}$, las cuales se concatenan de la forma

Figura 12. Ejemplo de *Word Embedding* sobre una frase $\mathbf{W}$. A cada palabra de la frase en español escrito se le asigna un *token* numérico. Al paso m de decodificación se concatenan las primeras m palabras, representadas como x_m, las cuales ingresan a una capa lineal con dimensión D que tiene como objetivo encontrar relaciones con respecto a la siguiente palabra. En este caso $D = 128$ y $M = 10$. La salida es un conjunto **E** de vectores embebidos con dimensión $M \times D$



($MH^{DT} = \text{concat}(\mathbf{O}_1^{DT}, \mathbf{O}_2^{DT} \ldots, \mathbf{O}_C^{DT}) W_{Mh^{DT}}$), según un ajuste de pesos $W_{Mh^{DT}}$ y se obtiene una única representación $\hat{\mathbf{E}}$ a través de dos capas lineales. Cabe destacar que se emplea una variación en el módulo MHA, dado a que en este caso es necesario utilizar una máscara sobre las matrices de atención, con el fin de que en el paso de decodificación m, el módulo solo pueda determinar relaciones entre palabras decodificadas hasta el paso $m - 1$, garantizando así un correcto modelamiento del lenguaje. De esta forma se puede contener las limitaciones en la fase de entrenamiento, en cuanto a las relaciones entre palabras ya decodificadas. Para ello, la máscara solo tiene en cuenta los elementos de la matriz triangular inferior, ponderando con 0 los demás elementos del conjunto. La figura 13 ilustra un ejemplo con las matrices de atención obtenidas en este módulo.

Figura 13. Mapas de atención de lengua escrita $\mathbf{A}^{DT}$. Nótese como los elementos por fuera de la matriz triangular inferior no poseen ninguna correlación con algún otro. Este ejemplo cuenta con ocho cabezas y dos capas de codificación. La información extraída corresponde a la segunda capa.



Hasta este punto, la información en $\hat{\mathbf{E}}$ contiene relaciones temporales de la secuencia de palabras $\mathbf{W}$, sin embargo, para realizar una traducción a nivel de lengua escrita es necesario encontrar relaciones entre el video y el texto $(\mathbf{F} \rightarrow \mathbf{W})$. En este sentido, un segundo módulo MHA encuentra proyecciones entre los dos dominios, explotando información temporal entre las palabras y la información codificada en el conjunto de embebidos $\hat{\mathbf{K}}$, producto del módulo codificador. Esta operación representa un proceso altamente importante, debido a que es la pieza en donde se aprenden relaciones representativas utilizando la información embebida de glosas. Para ello, las representaciones $\hat{\mathbf{H}}_i^{CT} = W_{\hat{h}^{CT}}\hat{\mathbf{K}}$, $\hat{\mathbf{V}}_i^{CT} = W_{\hat{v}^{CT}}\hat{\mathbf{K}}$ y $\hat{\mathbf{Q}}_i^{DT} = W_{\hat{q}^{DT}}\hat{\mathbf{E}}$, determinan un conjunto de características $\hat{\mathbf{O}}^{DT}$, donde a partir de los mapas de atención $\hat{\mathbf{A}}_i^{DT} = \sigma((\hat{\mathbf{H}}_i^{CT})^T\hat{\mathbf{Q}}_i^{DT})$ y la concatenación de las características $\hat{\mathbf{O}}_i^{DT} = \hat{\mathbf{A}}_i^{DT}\hat{\mathbf{V}}_i^{CT}$;

$\hat{\mathbf{O}}_i^{DT} \in \mathbb{N}^{M \times \frac{d}{C}}$ se obtiene una única representación $\tilde{\mathbf{E}} \in \mathbb{R}^{M \times D}$.

El conjunto de embebidos resultante $\tilde{\mathbf{E}}$ se da como entrada a una capa lineal con activación *softmax*, donde la dimensionalidad del tensor se ajusta acorde al tamaño del diccionario de palabras escritas y se obtiene una distribución de probabilidad por cada palabra, obteniendo así $P(w_m|F, G, w_{1:m-1})$. Además, se debe tener en cuenta que la decodificación incluye dos módulos de múltiple atención, los cuales constituyen una única capa de procesamiento B_x^{DT}.

En cuanto al entrenamiento del decodificador se utilizó la función de pérdida *cross entropy*. Para ello, primero se calcula la probabilidad de las palabras dado el vídeo y las glosas, lo que se define como $p(W|\tilde{E}) = \prod_{m=1}^{M} p(w_m|\tilde{e}_m)$. Donde la pérdida se obtiene a partir de $\mathbb{L}_{DT} = 1 - \prod_{m=1}^{M} \sum_{j=1}^{J} P(w^{*j}_m) P(w^{*\,j}_m|\tilde{e}_m)$, donde J representa el tamaño del diccionario de palabras del lenguaje escrito y $P(w^{*j}_m)$ representa las probabilidades de las palabra objetivo w^{*j} al paso m de decodificación.

Finalmente, con el fin de obtener reconocimiento de glosas y traducción de LS en una sola estrategia, se utilizó un entrenamiento conjunto en el método propuesto. Para ello, se emplean las funciones de pérdida de los módulos CT ($\mathbb{L}_{CT}$) y DT ($\mathbb{L}_{DT}$), donde cada una está ponderada por una constante λ, la cual cuantifica la relevancia de cada pérdida en la etapa de entrenamiento. Por lo tanto, la función de pérdida general se define como: $\mathbb{L} = \lambda_{CT}\mathbb{L}_{CT} + = \lambda_{DT}\mathbb{L}_{DT}$.

5. DISEÑO EXPERIMENTAL

5.1. CONJUNTOS DE DATOS

El método propuesto fue ampliamente evaluado en dos diferentes conjuntos de datos dedicados a la traducción de la LS y el reconocimiento de glosas. Los dos conjuntos de datos son públicos y se describen a continuación:

5.1.1. CoL-SLTD: *Colombian Sign Language Dataset*. El conjunto de datos Col-SLTD contiene videos con señas reales y comunes de la LS colombiana, donde cuenta con la correspondiente traducción en lengua escrita y su representación en glosa. Este conjunto de datos representa el primer esfuerzo por cuantificar señas en video, obtener relaciones de glosas y registrar una amplia variabilidad de la región oriente del país. Las frases fueron realizadas por 13 intérpretes entre 21 y 80 años. Un total de 24 oraciones afirmativas, 4 negativas y 11 interrogativas hacen parte de este conjunto de datos; cada frase sigue una gramática establecida (sujeto, verbo, objeto) y fue repetida 3 veces por cada interprete; la figura 14 muestra fotogramas extraídos del conjunto de datos. Por otro lado, el conjunto de datos completo tiene un vocabulario de 114 palabras y 90 glosas, contando con un total de 1020 videos. Cada video tiene una resolución espacial de 448 $\times$ 448 y una resolución temporal de 30 fotogramas por segundo. Además, CoL-SLTD cuenta con diferentes representaciones de video, tales como: RGB, flujo óptico y estimación de la pose.

CoL-SLTD cuenta con dos divisiones de datos, las cuales buscan evaluar el modelo desde diferentes enfoques. La primera división evalúa la capacidad del método en identificar características visuales relevantes, esto se logra debido a que se usan las mismas frases tanto para entrenamiento como para testeo, variando entre etapas las personas que interpreta la frase. La segunda división tiene como fin evaluar la

capacidad de la red en modelar la LS, debido a que se usan los mismos intérpretes en ambas fases, variando entre etapas las frases de la LS. El cuadro 1 muestra la distribución de los datos en ambas divisiones. Este conjunto de datos fue aprobado por el Comité de Ética de la Universidad Industrial de Santander en Bucaramanga, Colombia, con el número $D19 - 13353$. Se obtuvo el consentimiento informado por escrito de cada participante.

Cuadro 1. Distribución de datos en CoL-SLTD.

	DIVISIÓN 1		**DIVISIÓN 2**	
	Train	*Test*	*Train*	*Test*
Número de vídeos	807	213	922	98
Número de intérpretes	10	3	13	13
Número de oraciones	39	39	35	4
Número de glosas	90	90	90	90
Número de palabras	110	110	110	16

Aumento de datos: Con el fin de obtener un conjunto de datos más representativo en la etapa de entrenamiento, un aumento de datos fue llevado a cabo sobre CoL-SLTD. Para ello, se emplearon estrategias de giro horizontal sobre los fotogramas de los videos, con el fin de simular los cambios de la mano dominante del intérprete.

5.1.2. RWTH-PHOENIX-Weather 2014 T. Con el fin de evaluar de forma exhaustiva el método propuesto, en este trabajo se utilizó uno de los conjuntos de datos más utilizado en el estado del arte y en los desafíos más grandes de visión por computador. El conjunto de datos se llama RWTH-PHOENIX-Weather 2014 T y corresponde a la LS alemana, donde 9 intérpretes representan las noticias del clima a través de este lenguaje. RWTH-PHOENIX-Weather 2014 T cuenta con un total de 2887 palabras, 1066 glosas y 8257 videos, donde los autores proponen un subconjuntos de 7096 videos para entrenamiento, 642 para testeo y 519 para desarrollo. El cuadro 2 muestra de forma más detallada la distribución de datos.

Este conjunto de datos es ampliamente utilizado en diferentes estrategias de tra-

Figura 14. Ejemplos de fotogramas del conjunto de datos CoL-SLTD. Cada video fue grabado bajo un entorno controlado, utilizando como fondo una pantalla verde y condiciones de luz ideales. La posición de los intérpretes es de forma frontal a la cámara, usando ropa de diferente color al fondo.

Cuadro 2. Distribución de datos en RWTH-PHOENIX-Weather 2014 T.

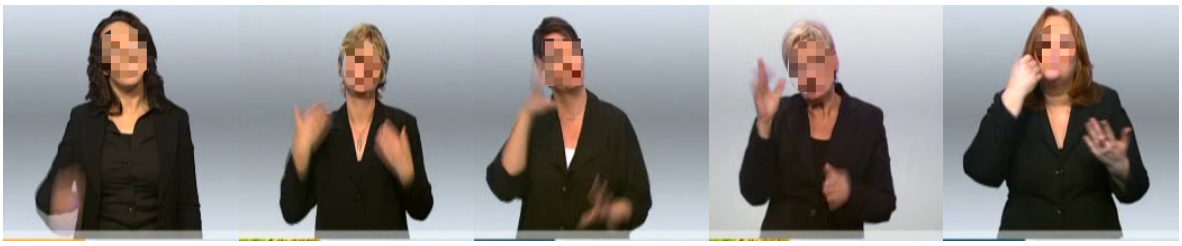
	Train	**Dev**	**Test**
Número de videos	7096	519	642
Número de palabras	2887	951	1001
Número de glosas	1066	393	411

ducción de LS, debido a que cuenta con una alta riqueza informativa, lo cual lo convierte en uno de los conjuntos más robustos para este problema. La figura 15 muestra fotogramas extraídos del conjunto de datos.

5.2. CONFIGURACIÓN DE LA ESTRATEGIA

5.2.1. Configuración del modelo. Para la validación y evaluación, el modelo propuesto fue fijado con parámetros de funcionamiento, los cuales se mantuvieron durante todos los experimentos reportados. Primero, se determinó un tamaño de resolución y un número de fotogramas para cada video, el cual corresponde a (128 $\times$ 227 $\times$ 227 $\times$ 3), donde sus dimensiones a través de la red se ven afectadas entre

Figura 15. Ejemplos de fotogramas del conjunto de datos RWTH-PHOENIX-Weather 2014 T. Cada video fue grabado sobre un fondo gris, donde cada intérprete viste ropa de color negro.



cada módulo. En el cuadro 3 se especifican como se ven afectados los tensores de representación en cada una de las etapas de la estrategia propuesta y como se baja la dimensionalidad de los videos de entrada a los descriptores embebidos correspondientes, los cuales entran en relación directa con las secuencias generadas. Durante la validación se tuvo en cuenta una arquitectura simétrica (el mismo número de capas B_x para el codificador y decodificador) y se hicieron pruebas con 1, 2, 3 y 4 capas.

Cuadro 3. Dimensionalidad del tensor a través del método propuesto. La dimensión inicial del tensor de video corresponde a $(T \times W \times H \times C)$. El módulo CT cuenta con dos salidas, debido a que una ingresa al módulo CTC y otra al módulo DT.

Módulo	**Dimensión de salida**
Capa de entrada de $\mathbf{F}$	(120 $\times$ 227 $\times$ 227 $\times$ 3)
ECET	(58 $\times$ 128)
CT	(58 $\times$ 128), (58 $\times$ 91)
Capa de entrada de $\mathbf{W}$	(12)
Word embedding	(12 $\times$ 128)
DT	(12 $\times$ 115)

En el cuadro 4 se resume la arquitectura propuesta y los hiperparámetros utilizados en la red convolucional volumétrica, la cual resulta primordial en este trabajo para encontrar una correspondencia cinemática, según las secuencias de video y flujo óptico. Esta estrategia permite obtener descriptores complejos, que resumen la in-

formación observada en las secuencias $2d + t$. En esta arquitectura se inician con secuencias de video en tres canales, o representaciones de flujo en cuanto a sus direcciones y magnitud. Esta entrada es proyectada a través de capas que implementan convoluciones 3D locales, los cuales van incrementando la profundidad de canales, mientras se reduce la dimensionalidad espacial. Finalmente, la arquitectura permite obtener bloques embebidos que representan las secuencias de entrada.

Cuadro 4. Dimensionalidad del tensor a través del módulo extractor de características espacio-temporales (ECET). Los parámetros para las operaciones de capas *max pooling 3D* correspoden a un *pool size* de $(3 \times 3 \times 3)$ y un tamaño de *stride* de $(2 \times 2 \times 2)$.

Capa	**Tamaño de *stride***	**Tamaño de *kernel***	**Dimensión de salida**
Entrada	-	-	(128, 227, 227, 3)
Bloque 1	(2, 2, 2)	(3, 3, 3)	(63, 57, 57, 64)
Bloque 2	(3, 3, 3)	(1, 1, 1)	(62, 28, 28, 32)
Bloque 3	(3, 3, 3)	(1, 1, 1)	(61, 14, 14, 64)
Bloque 4	(3, 3, 3)	(1, 1, 1)	(60, 7, 7, 64)
Bloque 5	(3, 3, 3)	(1, 1, 1)	(59, 3, 3, 128)
Bloque 6	(3, 3, 3)	(1, 1, 1)	(58, 1, 1, 128)
Redimensión	-	-	(58, 128)

Durante el entrenamiento, también se fijaron hiperparámetros en la arquitectura entrenada de extremo a extremo. El cuadro 5 muestra un resumen de los hiperparámetros utilizados para la etapa de entrenamiento.

Cuadro 5. Hiperparámetros utilizados en el método propuesto. Por otro lado, luego de la quinta época, la tasa de aprendizaje disminuye acorde a: $lr = lr * e^{-0,1}$.

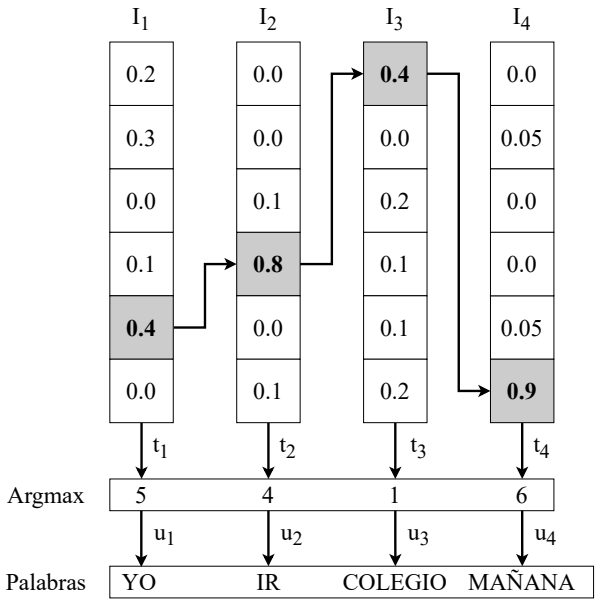
Hiperparámetro	**Valor**
Tamaño de *batch*	1
Número de neuronas en las capas *feed forward*	2048
Dropout en módulos MHA	0,1
Número de épocas de entrenamiento	30
Optimizador	Adam
Tasa de aprendizaje	1 $\times 10^{-4}$

Por otra parte, para obtener estas unidades escritas con respecto a las glosas ob-

servadas, la metodología propuesta determina $P(G|F)$ y $P(W|F,G)$. Para ello es necesario utilizar métodos de decodificación que permitan la transformación de una distribución probabilística en una representación a nivel de palabra. Debido a esto, los métodos *greedy search* y *beam search* fueron utilizados en este trabajo, donde ambos tienen como objetivo determinar una secuencia de palabras $\mathbf{U} = (u_1, u_2, \ldots, u_M)$, dado un conjunto de probabilidades $\mathbf{I} \in \mathbb{R}^{M \times D}$, donde M representa el número de palabras y D el tamaño del diccionario de palabras. A continuación, se describen de manera específica los dos métodos:

- **Greedy search** es un método básico que en cada paso de decodificación determina una representación escrita u_m con base en la palabra más probable. Como se observa en la figura 16, el algoritmo determina un token t_m con base en $t_m = Argmax(I_m)$, donde posteriormente se obtiene la palabra u_m asociada a t_m.

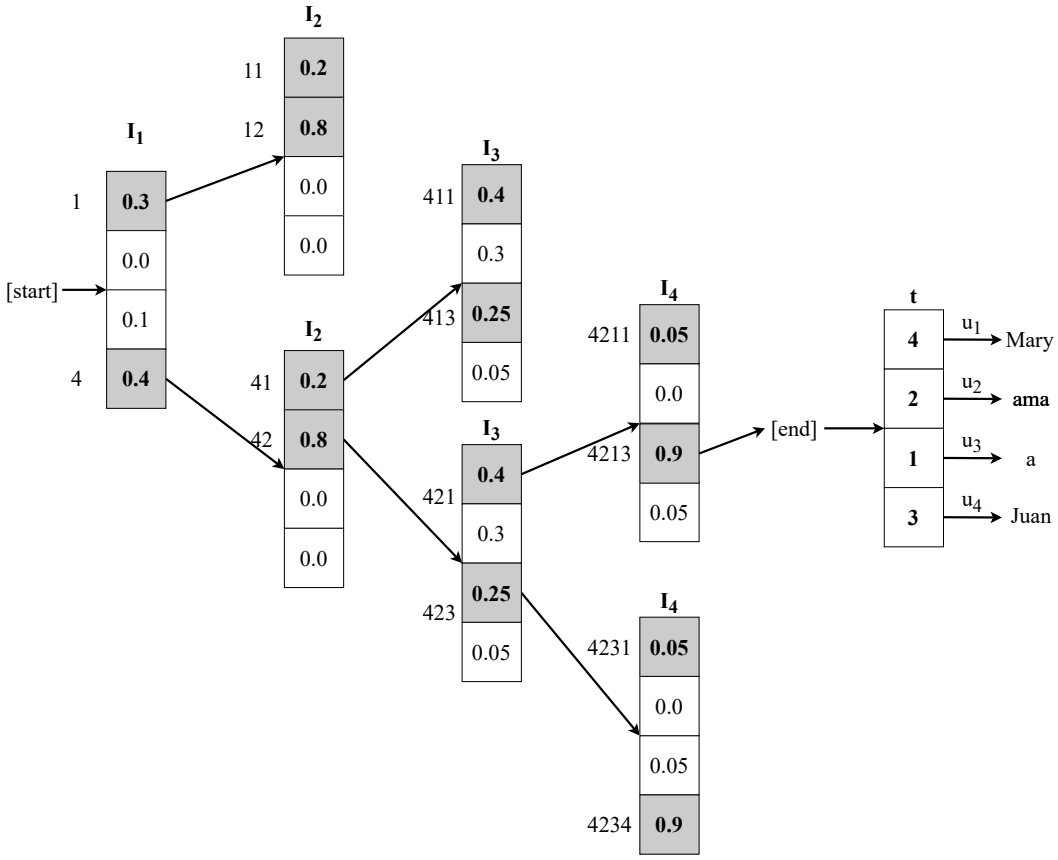
Figura 16. Ejemplo de decodificación utilizando *greedy search*.



- **Beam search** se basa en diferentes caminos de hipótesis, buscando obtener una secuencia $\mathbf{U}$ de palabras con base en el camino más probable. Como

se observa en la figura 17, usando un árbol de probabilidades, en cada paso m de decodificación se determinan los β elementos con los valores más altos, expandiendo progresivamente los niveles del árbol [22]. Asimismo, en cada paso m se ponderan las probabilidades de los caminos obtenidos, descartando en cada paso los menos probables. Finalmente, se obtiene una secuencia t_m de *tokens* que representan la frase decodificada, donde la secuencia $\mathbf{U}$ se obtiene con base en la palabra asociada a cada *token*.

Figura 17. Ejemplo de decodificación utilizando *beam search*. El algoritmo en cada paso m de decodificación determina los β elementos más probables, en este caso $\beta = 2$.



[22] P Hayes-Roth y col. "Speech understanding systems: Summary of results of the five-year research effort". En: (1976).

5.3. VALIDACIÓN ESTADÍSTICA

El método propuesto fue validado con respecto a los dos conjuntos de datos públicos descritos en la sección anterior (ver cuadro 1 y cuadro 2). Para ello, se utilizó una única partición de entrenamiento y evaluación (*train-test*), según fueron establecidas por los equipos que publicaron los datos, lo que facilita su comparación con el estado del arte. Además, el objetivo principal de este proyecto consistió en realizar traducciones entre secuencias de gestos y su equivalencia escrita, pero apoyado en representaciones intermedias, como son las secuencias de glosas. Es así como para el trabajo propuesto resulta determinante establecer el aporte de las glosas en la tarea de traducción. Con el fin de cuantificar el aporte de esta representación intermedia, fue necesario implementar una variación de la estrategia, la cual tiene como objetivo realizar traducciones directas entre el video y el texto (denominada *sign2text*).

En cuanto a las métricas de validación, en este trabajo se utilizaron medidas entre secuencias, las cuales han sido ampliamente utilizadas para validar estrategias de traducción. A continuación, se detallan las métricas utilizadas:

- **La Métrica de evaluación *Word Error Rate* (WER)** toma en cuenta el número de palabras sustituidas (S), eliminadas (E) e insertadas (I) entre una frase de hipótesis y una frase de referencia. Para ello, se utiliza la formula $WER = \frac{S+E+I}{N}$, donde N representa el número total de palabras de la frase de referencia. Por ejemplo, teniendo la frase de referencia “YO NECESITAR NUEVA COMPUTADORA” y la frase de hipótesis “NECESITAR MAÑANA GRANDE COMPUTADORA”, se tiene un total de 1 eliminación (la glosa “YO”), 1 sustitución (la glosa “NUEVA” por “MAÑANA”) y 1 inserción (la glosa “GRANDE”). En este caso, existen 3 errores sobre un total de 4 palabras de la frase de referencia, por lo tanto, el WER corresponde a un 75 %.

- **Métrica de evaluación *Bilingual Evaluation Understudy* (BLEU)** es comúnmente utilizada en problemas de traducción entre dos lenguajes, midiendo niveles de correspondencia entre n-gramas [23]. Como se muestra en el ejemplo de el cuadro 6, generalmente se utilizan conjuntos de hasta 4 n-gramas en ambas frases, donde se verifica si los elementos de cada conjunto de la frase hipótesis se encuentran en el conjunto de la frase de referencia. Los indicadores BLEU-1 y BLEU-2 evalúan el rendimiento de la red en secuencias cortas de palabras, mientras que los indicadores BLEU-3 y BLEU-4 tienen en cuenta secuencias más largas. Por lo tanto, el indicador que mejor representa la calidad de traducción es el BLEU-4, ya que está sujeto a la mayor secuencia de n-gramas.

Cuadro 6. Ejemplo de métrica BLEU entre dos frases. En este caso, la frase de referencia corresponde a "yo necesito una computadora nueva", mientras que la frase de hipótesis es "yo necesito una nueva". Como se puede observar, las frases se dividen acorde al número de n-gramas, donde se hace una comparación entre conjuntos y se comprueba la cantidad de elementos en común entre la frase hipótesis y la frase de referencia.

n	**Conjunto de n-gramas de la frase referencia**	**Conjunto de n-gramas de la frase hipótesis**	**BLEU**
1	yo, necesito, una, computadora, nueva	yo, necesito, una, nueva	80 %
2	yo necesito, necesito una, una computadora, computadora nueva	yo necesito, necesito una, una nueva	50 %
3	yo necesito una, necesito una computadora, una computadora nueva	yo necesito una, necesito una nueva	33,33 %
4	yo necesito una computadora, necesito una computadora nueva	yo necesito una nueva	0 %

[23] Kishore Papineni y col. "Bleu: a method for automatic evaluation of machine translation". En: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, págs. 311-318.

6. EVALUACIÓN Y RESULTADOS

El método propuesto fue exhaustivamente evaluado con los conjuntos de datos: CoL-SLTD y RWTH-PHOENIX-WEATHER 2014 T. A continuación, se muestran los resultados obtenidos, ordenados por cada conjunto de datos.

6.1. RESULTADOS EN COL-SLTD

En cuanto a la validación sobre el conjunto de datos COL-SLTD se consideraron las dos divisiones de datos propuestas por los autores. La división 1 tiene frases similares en el entrenamiento y validación, pero utiliza diferentes intérpretes en cada partición. De esta manera se busca validar la capacidad de las estrategias para generalizar la representación de las secuencias, descritas por diferentes actores. En cuanto a la división 2, el conjunto de datos fue particionado también en entrenamiento y validación, pero utilizando frases diferentes en ambas particiones. Claramente esta partición es más desafiante, pretendiendo buscar que la red genere coherencia en el lenguaje sobre representaciones no observadas. Por otro lado, los resultados que se muestran a continuación corresponden a los obtenidos por el método de decodificación *beam search*, debido a que es el método con el que se obtuvieron las mejores métricas.

En este trabajo primero se realizó un estudio de parámetros con los principales componentes de la arquitectura, con el fin validar las condiciones para lograr el mejor desempeño. En este sentido, se validó el comportamiento de la arquitectura propuesta en cuanto al número de mecanismos de atención en cada módulo (número de cabezas), el número de capas sucesivas de procesamiento B en el *transformer* y también la ganancia al procesar patrones desde campos vectoriales de movimiento (flujo $(\mathbf{F})$) con respecto a secuencias crudas (RGB$(\mathbf{V})$). En este sentido, el primer

experimento consistió en validar la arquitectura propuesta variando el número C de cabezas utilizadas en los módulos múltiples de atención. En este experimento se fijaron las capas del *transformer* en $B = 2$ tanto para el codificador como el decodificador. El cuadro 7 resume los resultados obtenidos para las dos divisiones del conjunto de datos considerando las diferentes métricas de evaluación. Como se puede observar, el número de cabezas impacta de forma directa en la representación propuesta, obteniendo por ejemplo para la división 1 un BLEU-4 con variaciones entre $[59,26\,\% - 72,64\,\%]$ y un WER entre $[39,69\,\% - 50,1\,\%]$. Desde estos resultados se puede ver que un número de $C = 8$ cabezas logra el mejor desempeño, mientras para la tarea más desafiante de generación de frases (división 2) se requiere un mayor número de cabezas ($C = 16$) que permitan una mayor generalización del aprendizaje. En la segunda división, claramente existe un WER significativamente alto, sin diferencias significativas entre cabezas, hecho asociado a la complejidad de la representación de las frases. Con base en lo anterior, los siguientes experimentos contarán con un número fijo $C = 8$ para la división 1 y $C = 16$ para la división 2, ya que estos valores corresponden a los mejores resultados de traducción.

Cuadro 7. Análisis de la variable C sobre el método propuesto. El número de capas *transformer* corresponde a $B = 2$.

DIVISIÓN 1					
C	**WER**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
2	50,1	67,05	62,51	60,4	59,26
4	44,55	69,25	65,69	64,34	63,83
8	42,39	**77,53**	**74,42**	**73,16**	**72,64**
16	**39,69**	75,75	72,57	71,43	70,97
DIVISIÓN 2					
2	**81,88**	33,43	14,97	10,1	7,45
4	85,54	23,73	5,98	2,58	1,55
8	82,9	30,73	15,62	10,37	7,52
16	83,50	**34,53**	**16,21**	**11,27**	**8,24**

En un segundo experimento se validó la capacidad del método propuesto en cuanto

al número de capas B en el *transformer*. Para los experimentos, se consideró una arquitectura *transformer* simétrica con igual número de capas tanto en el codificador como en el decodificador. En este caso, una capa B_x^{CT} constituye un módulo de múltiples capas auto atencionales junto con diferentes módulos de procesamiento, mientras que una capa B_x^{DT} constituye una asociación de dos módulos múltiples de atención acompañados de distintas capas que contribuyen a la decodificación. El cuadro 8 muestra los resultados obtenidos en las tareas de reconocimiento y traducción en la división 1 de datos, donde se representa el desempeño de la metodología variando el número B de capas *transformer*, utilizando secuencias de movimiento aparente y secuencias crudas.

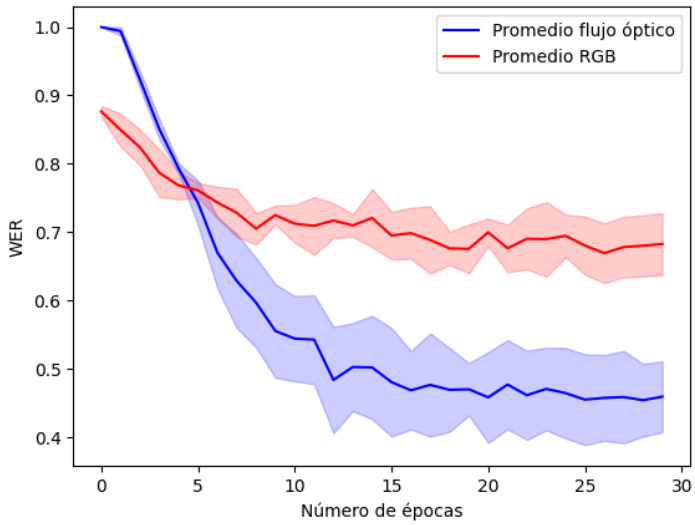
Cuadro 8. Comparación entre las representaciones de vídeo RGB y flujo óptico en la división 1 de CoL-SLTD. El valor de B corresponde al número de capas *transformer* utilizadas en el experimento. Se utilizaron 8 cabezas para los módulos *MHA*.

B	Representación	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	RGB (**V**)	71,81	42,35	35,94	33,45	32,65
	Flujo (**F**)	**48,09**	**72,44**	**67,7**	**65,43**	**64,34**
2	RGB (**V**)	63,62	50,82	43,83	41,05	39,92
	Flujo (**F**)	**42,39**	**77,53**	**74,42**	**73,16**	**72,64**
3	RGB (**V**)	64,97	51,69	44,01	40,97	39,75
	Flujo (**F**)	**35,42**	**74,65**	**71,58**	**70,48**	**69,95**
4	RGB (**V**)	61,85	47,75	41,85	39,71	38,98
	Flujo (**F**)	**38,67**	**75,33**	**72,57**	**71,58**	**71,26**

Como se esperaba, las secuencias de movimiento aparente (flujo $(\mathbf{F})$) tienen un impacto claro en la tarea de reconocimiento, representación y traducción de gestos en la LS. De hecho, en cualquier de las configuraciones se logra una ganancia superior al $20\,\%$, con respecto a la secuencia cruda de video. Estas diferencias son coherentes con la hipótesis planteada en el presente trabajo, mostrando que una representación cinemática, que describe además la geometría de las posturas, impacta claramente en la representación de los gestos, siendo menos invariante con factores relacionados con la captura de la información. De hecho, la representa-

ción obtiene un error de reconocimiento de cerca del $35\,\%$ y una alta calidad de traducción, donde el BLEU-4 más alto corresponde a un $72,64\,\%$. Asimismo, en la figura 18 se ilustra un ejemplo de la métrica WER en la etapa de entrenamiento para la división 1, mostrando un notable mejor reconocimiento de glosas en todos los experimentos con flujo óptico, donde esta representación disminuye de manera drástica el error en las primeras épocas de entrenamiento, garantizando un modelado confiable al finalizar esta etapa. Además, la representación en RGB muestra un estancamiento entre épocas, ya que presenta diferencias poco relevantes en cada iteración.

Figura 18. Comparación de la métrica WER en la división 1 de CoL-SLTD a través de las épocas de entrenamiento entre las representaciones RGB y flujo óptico.



Por otra parte, la modificación de capas en el *transformer* no tiene una contribución marcada, permitiendo un número reducido de capas $B = 2$ para obtener resultados coherentes, con una buena relación con respecto a los parámetros necesarios en la configuración computacional. Este hecho puede estar asociado a la dependencia del número de datos que se tienen para el entrenamiento, los cuales pueden no ser suficientes para el aprendizaje de un elevado número de parámetros que se incre-

mentan con el número de capas y pueden desvanecer la representación. En este sentido, a partir de una representación en flujo óptico, una metodología compacta modela de mejor forma las características de la LS. Asimismo, el mejor reconocimiento de glosas corresponde al caso de $B = 3$, donde el flujo óptico presenta un WER menor en un $29,55\,\%$ a comparación de la representación RGB.

Esta misma validación fue realizada sobre el conjunto de datos en la división 2. El cuadro 9 resume el desempeño obtenido por el método propuesto utilizando diferentes secuencias de entrada y haciendo variaciones en el número de capas. De forma similar, la mayoría de los mejores resultados se obtuvieron utilizando el flujo óptico como representación de entrada. Sin embargo, en este caso no existe una diferencia significativa entre representaciones, donde inclusive en algunos casos los videos en RGB obtuvieron resultados considerables. En este caso, la mejor configuración fue lograda con $B = 4$ capas y utilizando secuencias de movimiento aparente, logrando un $BLEU - 4 = 14,64$ para generar textos coherentes que no han sido observados. Cabe destacar que estos resultados son significativamente inferiores a los obtenidos en la primera división, debido a que en esta tarea se generan frases que no han sido previamente observadas en las secuencias de entrenamiento.

Cuadro 9. Comparación entre las representaciones de vídeo RGB y flujo óptico en la división 2 de CoL-SLTD. El valor de B corresponde al número de capas *transformer* utilizadas en el experimento. Se utilizaron 16 cabezas para los módulos *MHA*.

B	Representación	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	RGB (V)	92,09	9,07	0	0	0
	Flujo (F)	**75,17**	**36,73**	**16,02**	**10,18**	**6,96**
2	RGB (V)	84,78	30,47	11,96	7,79	5,68
	Flujo (F)	**83,5**	**34,53**	**16,21**	**11,28**	**8,24**
3	RGB (V)	91,83	**28,05**	8,26	4,3	2,64
	Flujo (F)	**79,08**	26,75	**8,38**	**4,87**	**2,91**
4	RGB (V)	88,45	35,16	14	9,29	6,59
	Flujo (F)	**83,41**	**41,01**	**25,33**	**19,36**	**14,64**

Uno de los principales intereses de este trabajo es determinar la contribución de las

glosas como representación intermedia para apoyar la LS. Es por ello que en un tercer experimento se cuantificó el aporte de la representación intermedia de glosas. Por lo tanto, se implementó una variación de la arquitectura propuesta (denominada *sign2text*), la cual elimina el módulo RG, pieza principal para el reconocimiento de la representación en glosa, donde el objetivo de esta variación es realizar traducciones directas de video a texto. Por parte de la división 1, el cuadro 10 muestra que de forma notable, los mejores resultados en todos los experimentos corresponden al método propuesto, contando con una importante superioridad en cuanto a la calidad de traducción frente a la variación *sign2text*. Específicamente, el método propuesto obtiene un BLEU superior en todos los conjuntos de n-gramas, donde la métrica BLEU-4 varía entre $[64,34\,\% - 72,64\,\%]$, superando al método *sign2text* en hasta en un $61,96\,\%$. Asimismo, se observa que a medida que el número de capas B va aumentando, el método propuesto presenta métricas similares en todos los casos, donde con un $B = 2$ se obtiene la mejor traducción y un $B = 3$ se obtiene el mejor reconocimiento de glosas. De forma interesante, en la metodología *sign2text*, a medida que el modelo es más profundo, los resultados presentan una decadencia, esto se puede atribuir a que no existen relaciones temporales entre la LS y un lenguaje escrito, por lo tanto, el método en lugar de explotar proyecciones entre los dos dominios, se le dificulta encontrar las relaciones no lineales entre el video y el texto, ya que son inexistentes.

Por otro lado, como se muestra en el cuadro 11, los resultados de los experimentos sobre la división 2 siguen la tendencia de la división 1, dado que en todos los casos las mejores métricas corresponden al método propuesto. De forma específica, el error en reconocimiento de glosas varía entre $[75,17\,\% - 83,5\,\%]$ y el BLEU-4 se encuentra en los rangos de $[2,91\,\% - 14,64\,\%]$. Cabe resaltar que en la mayoría de los experimentos sobre el método *sign2text* se obtienen métricas de traducción de cero, indicando que para la metodología es más complejo determinar frases no

Cuadro 10. Comparación entre el método propuesto y la variación *sign2text* en la división 1 de Col-SLTD. El valor de B corresponde al número de capas *transformer* utilizadas en el experimento. Se utilizaron 8 cabezas para los módulos *MHA*.

B	Método	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	Método *sign2text*	-	60,61	55	52,72	51,51
	Método propuesto	**48,09**	**72,44**	**67,7**	**65,43**	**64,34**
2	Método *sign2text*	-	50,44	44,34	41,81	40,28
	Método propuesto	**42,39**	**77,53**	**74,42**	**73,16**	**72,64**
3	Método *sign2text*	-	39,49	33,38	30,46	29,03
	Método propuesto	**35,42**	**74,65**	**71,58**	**70,48**	**69,95**
4	Método *sign2text*	-	19,37	13,79	10,64	9,3
	Método propuesto	**38,67**	**75,33**	**72,57**	**71,58**	**71,26**

vistas durante el entrenamiento sin una representación intermedia de glosas.

Cuadro 11. Comparación entre el método propuesto y la variación *sign2text* en la división 2 de Col-SLTD. El valor de B corresponde al número de capas *transformer* utilizadas en el experimento. Se utilizaron 16 cabezas para los módulos *MHA*.

B	Método	WER	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	Método *sign2text*	-	15,26	0	0	0
	Método propuesto	**75,17**	**36,73**	**16,02**	**10,18**	**6,96**
2	Método *sign2text*	-	18,23	4,15	0	0
	Método propuesto	**83,5**	**34,53**	**16,21**	**11,27**	**8,24**
3	Método *sign2text*	-	18,11	3,71	0	0
	Método propuesto	**79,08**	**26,75**	**8,38**	**4,87**	**2,91**
4	Método *sign2text*	-	9,07	0	0	0
	Método propuesto	**83,41**	**41,01**	**25,33**	**19,36**	**14,64**

Con base en los resultados de estos experimentos, la estrategia propuesta tiene la capacidad de realizar traducciones más efectivas en secuencias cortas y largas de n-gramas utilizando información de glosas, por lo tanto, estos resultados muestran la importancia del modelamiento de glosas como paso intermedio en la traducción de la LS.

Un cuarto conjunto de experimentos se realizó con el fin de comprobar el impacto en la traducción en lengua escrita al cambiar el valor del regularizador λ_{CT} para la etapa

de entrenamiento. Al aumentar este valor, se otorga más relevancia al error obtenido en el reconocimiento de glosas, por lo tanto, este experimento busca cuantificar la correlación que existe entre el modelado de glosas y la traducción final.

Cuadro 12. Resultados de experimentos variando el regularizador λ_{CT}. En ambos casos se utilizó la configuración experimental que presente mejores resultados de traducción. En el caso de la división 1, se utilizó un $B = 2$ y un $C = 8$, por otro lado, en la división 2 se empleó un $B = 4$ y un $C = 16$.

DIVISIÓN 1					
λ_{CT}	**WER**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
1	42,39	**77,53**	**74,42**	**73,16**	**72,64**
5	41,24	73,6	69,91	68,39	67,74
10	**40,33**	73,21	70,11	68,77	68,23
DIVISIÓN 2					
1	83,41	**41,01**	**25,33**	**19,36**	**14,64**
5	**78,65**	38,54	21,77	15,44	11,27
10	83,84	34,64	19,37	13,72	9,78

Como se aprecia en el cuadro 12, en la división 1 existe una relación inversamente proporcional entre el valor de λ_{CT} y la métrica WER, lo cual indica que se obtiene un mejor reconocimiento de glosas al ponderar con un valor más alto el error del CTC. Sin embargo, la disminución del error en el reconocimiento de glosas no proporciona una mejor traducción, donde curiosamente los mejores resultados se obtienen con base en el error más alto. No obstante, la variabilidad en el BLEU-4 no es alta, dado que este valor se encuentra entre $[68,23\,\% - 72,64\,\%]$. Por otro lado, en la división 2 no se aprecia una relación entre λ_{CT} y la métrica WER, donde esta varía entre $[78,65\,\% - 83,84\,\%]$, y además, no se presentan mejores resultados de traducción.
Asimismo, la estrategia propuesta fue comparada con otros métodos del estado del arte, valorados en el mismo conjunto de datos. El cuadro 13 muestra una comparación entre métodos del estado del arte y la estrategia implementada. El método NSLT hace referencia a los resultados del modelo [3] mostrados en [7], mientras que el método RFSV corresponde a la estrategia planteada en [7]. En ambas divisiones

del conjunto de datos, el método propuesto superó las métricas del estado del arte, obteniendo un $8,41\,\%$ de superioridad en la métrica BLEU-4 en la división 1 y superando en un $5,95\,\%$ el BLEU-4 en la división 2, representando una mayor calidad de traducción de la LS desde diferentes enfoques evaluativos. Los bajos resultados en la división 2 se pueden atribuir a la naturaleza de las metodologías secuencia-secuencia, dado que en la etapa de entrenamiento se emplea la técnica de *teacher forcing* [24] durante cada paso de decodificación, por lo tanto, para la red es complejo determinar frases no vistas durante el entrenamiento.

Cuadro 13. Comparación entre el método propuesto y métodos del estado del arte sobre el conjunto de datos CoL-SLTD.

DIVISIÓN 1					
Método	**WER**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
NSLT	-	71,58	67,27	65,23	64,23
RFSV	-	47,8	40,44	37,39	35,81
Método *sign2text*	-	60,61	55	52,72	51,51
Método propuesto	**35,42**	**77,53**	**74,42**	**73,16**	**72,64**
DIVISIÓN 2					
NSLT	-	39,67	18,94	12,17	8,69
RFSV	-	30,05	12,86	7,09	4,65
Método *sign2text*	-	18,23	4,15	0	0
Método propuesto	**75,17**	**41,01**	**25,33**	**19,36**	**14,64**

Finalmente, de modo de ilustración, el cuadro 14 muestra algunas frases traducidas por el método propuesto; como se puede apreciar, la estrategia es capaz de reconocer glosas y realizar traducciones de calidad. Cabe destacar que la traducción en lengua escrita está altamente basada en el reconocimiento de las glosas, un claro ejemplo es la frase predicha “¿Tú cuantos años tienes”, a la cual, la secuencia de glosas reconocidas corresponde a “TÚ QUE AÑOS?”, sin embargo, la frase que se

[24] Alex M Lamb y col. “Professor forcing: A new algorithm for training recurrent networks”. En: *Advances in neural information processing systems* 29 (2016).

deseaba obtener corresponde a "¿Tú que haces?". Esto muestra que el método propuesto toma de forma relevante la información extraída a nivel de glosas, utilizándola como guía para obtener una traducción final.

Cuadro 14. Ejemplos de frases traducidas por el método propuesto en el conjunto de datos CoL-SLTD. En este caso, los ejemplos mostrados corresponden a traducciones obtenidas en los datos de testeo de la división 1.

	Frase real	**Frase predicha**
Glosa	CARLOS VIAJAR BOGOTÁ HOY	CARLOS VIAJAR BOGOTÁ HOY
Lengua escrita	Carlos viaja a Bogotá hoy.	Carlos viaja a Bogotá hoy.
Glosa	JUAN GUSTAR ESTO Y ESO	JUAN GUSTAR ESTO Y
Lengua escrita	A Juan le gusta esto y eso.	A Juan le gusta esto y eso.
Glosa	ELLA TRAER PERRO COLEGIO MAÑANA	ELLA TRAER COLEGIO MAÑANA
Lengua escrita	Ella va a traer un perro al colegio mañana.	Ella va a traer un perro al colegio mañana.
Glosa	TÚ QUE HACER?	TÚ QUE AÑOS?
Lengua escrita	¿Tú que haces?	¿Tú cuantos años tienes?

6.2. RESULTADOS EN RWTH-PHOENIX-WEATHER 2014 T

Una vez valorada la estrategia propuesta sobre el primer conjunto de datos, los ajustes valorados en la etapa de validación fueron transferidos para evaluar el desempeño en el conjunto de datos RWTH-PHOENIX-WEATHER 2014 T. Este conjunto de datos es mucho más grande y por lo tanto la fase de entrenamiento de altamente costosa, por lo que se reportó un experimento cuantitativo, el cual busca comparar la efectividad del método propuesto con otras metodologías del estado del arte. Para ello, se utilizó la mejor configuración experimental obtenida en los anteriores experimentos, los cuales corresponden a una representación de video en flujo óptico, un número de capas *transformer* $B = 2$ y un número de cabezas $C = 8$. El cuadro

15 muestra los resultados obtenidos en las tareas de reconocimiento y traducción, donde el método SLTT hace referencia a la estrategia propuesta en [5]. Como se puede apreciar, los resultados del método propuesto son inferiores en un $10,8\,\%$ en las métricas de traducción y un $62,59\,\%$ en la tarea de reconocimiento. Sin embargo, la estrategia propuesta es computacionalmente más compacta, reporta un menor número de parámetros, lo cual puede ser clave para hacer un despliegue en ambientes reales. También, la estrategia propuesta en este sentido puede adoptar y generalizar mejor la representación de la LS. De hecho, las métricas obtenidas representan un notable BLEU-4 de $11,58\,\%$, lo cual indica que una estrategia compacta puede generalizar de forma potencial la LS.

Cuadro 15. Comparación entre el método propuesto y métodos del estado del arte sobre el conjunto de datos RWTH-PHOENIX-WEATHER 2014 T.

TEST					
Método	**WER**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
NSLT	-	43,29	30,39	22,82	18,13
SLTT	**26,16**	**46,61**	**33,73**	**26,19**	**21,32**
Método propuesto	89,62	30,55	20,62	15,09	11,91
DEV					
NSLT	-	42,88	30,30	23,02	18,40
SLTT	**24,98**	**47,26**	**34,40**	**27,05**	**22,38**
Método propuesto	87,57	30,25	20,12	14,71	11,58

7. CONCLUSIONES Y PERSPECTIVAS

En este trabajo se presentó una arquitectura de tipo *transformer* que incluye múltiples mecanismos de atención para realizar la traducción automática desde segmentos de la LS, registrados en secuencias de video, hacia secuencias textuales. Una de las principales contribuciones en este trabajo fue la inclusión efectiva de las glosas como representación intermedia en las secuencias de video y la correspondencia textual en la lengua hablada. Como se evidenció en los resultados, sobre dos conjuntos de datos públicos, la representación intermedia de glosas aporta en el aprendizaje de representaciones efectivas de señas, así como también permite obtener arquitecturas que modelan de forma correcta la representación de la LS. En una segunda contribución del trabajo propuesto, las secuencias de video fueron procesadas como descriptores cinemáticos, primero calculando la secuencia correspondiente de movimiento aparente, la cual es proyectada a una arquitectura convolucional volumétrica. A partir de esta representación se demostró tener mayores fortalezas a las múltiples variaciones de fondo e iluminación, así como también permitió incluir componentes lingüísticos fundamentales en la LS, como son las primitivas cinemáticas. Con este método multitarea se logra un modelado robusto del lenguaje, realizando traducciones a nivel de lengua escrita y de glosas, las cuales son la única aproximación escrita que representa de manera correcta la LS.

El método propuesto fue evaluado exhaustivamente en dos conjuntos de datos públicos. En cuanto al conjunto de datos CoL-SLTD se demostraron altas capacidades para producir frases con amplia coherencia temporal cuando se tiene un conjunto de entrenamiento de las frases que serán predichas. También se lograron resultados notables, superando el estado del arte en tareas más desafiantes que pretendían medir la capacidad de la herramienta de producir nuevas frases que no son observadas en el conjunto de entrenamiento. Esto se ve evidenciado por los resultados

obtenidos en las métrica BLEU y WER, donde el método propuesto superó a las estrategias del estado del arte en ambas divisiones del conjunto de datos, obteniendo un BLEU-4 de $72,64\,\%$ y un WER de $35,42\,\%$ para la división 1 y un BLEU-4 de $14,64\,\%$ y un WER de $75,17\,\%$ para la división 2, indicando que el método es capaz de obtener traducciones eficaces sobre secuencias largas y cortas, reconocer gestos a través del tiempo de forma confiable y disminuir el sesgo a la hora de realizar traducciones ante frases no vista durante el entrenamiento.

Asimismo, los resultados obtenidos muestran que el uso de glosas como paso intermedio puede mejorar la traducción hasta en un $61,96\,\%$, lo que indica que el reconocimiento de esta representación impacta de forma relevante y positiva en la calidad de traducciones. Esta afirmación aplica para ambas divisiones del conjunto de datos CoL-SLTD, por lo tanto, la metodología propuesta presenta un mejor rendimiento desde diferentes perspectivas evaluativas, ya que tiene la capacidad de generalizar los gestos. Se concluye que el uso de glosas como paso intermedio en la traducción de LS contribuye notablemente al modelado de la lengua, resultando en traducciones mucho más confiables y coherentes. Por otro lado, en el presente trabajo se reafirmó que el uso de información de movimiento es una representación altamente significativa para el modelado de la lengua, ya que, al emplear información cinemática se obtuvieron traducciones superiores de hasta un $32,72\,\%$, esto se debe a que el movimiento es una característica natural e importante de la lengua.

Durante la validación, también la herramienta propuesta fue evaluada con el conjunto de datos RWTH-PHOENIX-WEATHER 2014 T, donde se obtuvo un competitivo BLEU-4 de $11,58\,\%$. La arquitectura logró resultados notables y capacidades de traducción, siendo competitivo con respecto a otras arquitecturas del estado del arte, que tienen una mayor dimensionalidad en los parámetros de aprendizaje. La estrategia propuesta en este sentido es compacta y robusta en cuanto a la capacidad de aprendizaje, utilizando un ajuste simple y una única transferencia de hiperparáme-

tros desde el otro conjunto de datos. Por lo tanto, el método propuesto puede ser fácilmente desplegado en aplicaciones reales, así como también puede ser reajustado con nuevas variabilidades del lenguaje.

El trabajo presentado pretendió dar soporte tecnológico a una de las principales brechas de comunicación de la población sorda. Además, su implementación y validación desde un conjunto de datos colombiano, tuvo como objetivo aproximar el modelamiento y la codificación de la LS colombiana. La arquitectura propuesta logra una representación geométrica de las gestos, que es a su vez enriquecida con información cinemática, obtenida de los campos vectoriales. También, los esquemas de múltiple atención logran aprender mapas de atención que representan una alternativa para lograr relaciones no lineales y alejadas en el tiempo para las secuencias codificadas. A pesar de los notables avances logrados con esta arquitectura, los resultados evidencian que se requieren diseñar nuevos mecanismos de aprendizaje que permitan lidiar con la amplia variabilidad del lenguaje, expresado tanto de forma textual, como las representaciones gesto-visuales. Además, se deben explorar mecanismos no supervisados que puedan generalizar las representaciones y logren capturar secuencias no vistas a partir de unidades mínimas de información. Por último, es necesario seguir avanzando en la recolección de datos, con el fin de obtener un conjunto de información más representativo a la LS colombiana, permitiendo mecanismos tecnológicos que puedan ser desplegados en ambientes reales.

BIBLIOGRAFÍA

Brox, Thomas y Jitendra Malik. “Large displacement optical flow: descriptor matching in variational motion estimation”. En: *IEEE transactions on pattern analysis and machine intelligence* 33.3 (2010), págs. 500-513 (vid. pág. 30).

Camgoz, Necati Cihan y col. “Neural sign language translation”. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, págs. 7784-7793 (vid. págs. 13, 22, 60).

Camgoz, Necati Cihan y col. “Sign language transformers: Joint end-to-end sign language recognition and translation”. En: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, págs. 10023-10033 (vid. págs. 13, 14, 24, 63).

Cheng, Ka Leong y col. “Fully convolutional networks for continuous sign language recognition”. En: *European Conference on Computer Vision*. Springer. 2020, págs. 697-714 (vid. pág. 23).

Cooper, Helen y col. “Sign language recognition using sub-units”. En: *The Journal of Machine Learning Research* 13.1 (2012), págs. 2205-2231 (vid. págs. 21, 22).

Cui, Runpeng, Hu Liu y Changshui Zhang. “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization”. En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, págs. 7361-7369 (vid. pág. 22).

De Coster, Mathieu, Mieke Van Herreweghe y Joni Dambre. “Sign language recognition with transformer networks”. En: *12th International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA). 2020, págs. 6018-6024 (vid. pág. 23).

Eberhard, David M., Gary F. Simons y Charles D. Fennig. *Ethnologue: Languages of the World*. 2022. URL: https://www.ethnologue.com/subgroups/sign-language (visitado 03-06-2022) (vid. pág. 12).

Hayes-Roth, P y col. “Speech understanding systems: Summary of results of the five-year research effort”. En: (1976) (vid. pág. 50).

Koller, Oscar, Jens Forster y Hermann Ney. “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers”. En: *Computer Vision and Image Understanding* 141 (2015), págs. 108-125 (vid. pág. 21).

Lamb, Alex M y col. “Professor forcing: A new algorithm for training recurrent networks”. En: *Advances in neural information processing systems* 29 (2016) (vid. pág. 61).

Massone, María Ignacia. “El habla visual: lingüística de las lenguas de señas”. En: *Signo y seña* 2 (1993), págs. 18-27 (vid. pág. 15).

Organization, World Health y col. “World report on hearing”. En: (2021), pág. 10 (vid. pág. 12).

Papineni, Kishore y col. “Bleu: a method for automatic evaluation of machine translation”. En: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, págs. 311-318 (vid. pág. 52).

Pigou, Lionel y col. “Sign language recognition using convolutional neural networks”. En: *European Conference on Computer Vision*. Springer. 2014, págs. 572-578 (vid. pág. 21).

Rodriguez, Jefferson y Fabio Martínez. “How important is motion in sign language translation?” En: *IET Computer Vision* 15.3 (2021), págs. 224-234 (vid. pág. 23).

Rodriguez, Jefferson y col. “Understanding Motion in Sign Language: A New Structured Translation Dataset”. En: *Proceedings of the Asian Conference on Computer Vision*. 2020 (vid. págs. 13, 60).

Sandler, Wendy. “The phonological organization of sign languages”. En: *Language and linguistics compass* 6.3 (2012), págs. 162-182 (vid. pág. 30).

Saunders, Ben, Necati Cihan Camgoz y Richard Bowden. “Progressive transformers for end-to-end sign language production”. En: *European Conference on Computer Vision*. Springer. 2020, págs. 687-705 (vid. pág. 24).

Supalla, Samuel J, Jody H Cripps y Andrew PJ Byrne. “Why American sign language gloss must matter”. En: *American annals of the deaf* 161.5 (2017), págs. 540-551 (vid. pág. 14).

Varol, Gül, Ivan Laptev y Cordelia Schmid. “Long-term temporal convolutions for action recognition”. En: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), págs. 1510-1517 (vid. pág. 32).

Vaswani, Ashish y col. “Attention is all you need”. En: *Advances in neural information processing systems*. 2017, págs. 5998-6008 (vid. págs. 13, 18, 34).

Yin, Kayo y Jesse Read. “Attention is all you sign: sign language translation with transformers”. En: *Proceedings of the European Conference on Computer Vision (ECCV) Workshop on Sign Language Recognition, Translation and Production (SLRTP)*. Vol. 23. 2020 (vid. pág. 24).

— “Better sign language translation with STMC-transformer”. En: *arXiv preprint arXiv:2004.00588* (2020) (vid. págs. 13, 24).