



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo

Instituto Tecnológico de Hermosillo

División de Estudios de Posgrado e Investigación

Traducción de Texto en Español a Texto LSM Usando Aprendizaje Profundo

TESIS

Presentada por:

Ing. Juan Carlos Hernández Cruz

Como requisito parcial para obtener el grado de
Maestro en Ciencias de la Computación

Director de Tesis:

M.C. César Enrique Rose Gómez

Codirector de Tesis:

Dr. Samuel González López

Hermosillo, Sonora, México

AGOSTO 2019



Av. Tecnológico S/N Colonia El Sahuaro, C.P. 83170
Hermosillo, Sonora. Tel.2606500 Ext. 136
e-mail: depi_hermosillo@tecnm.mx

www.tecnm.mx
www.ith.mx



ISO 9001:2015
Sistema de Gestión de Calidad Certificado



Instituto Tecnológico de Hermosillo

“2019, Año del Caudillo del Sur, Emiliano Zapata”

SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/247/19.
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS.

11 Septiembre de 2019

**C. JUAN CARLOS HERNÁNDEZ CRUZ,
PRESENTE.**

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre **“TRADUCCIÓN DE TEXTO EN ESPAÑOL A TEXTO LSM USANDO APRENDIZAJE PROFUNDO”**, que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

ATENTAMENTE

CÉSAR ENRIQUE ROSE GÓMEZ
DIRECTOR

MARÍA TRINIDAD SERNA ENCINAS
SECRETARIA

ANA LUISA MILLÁN CASTRO
VOCAL

ROSA IRENE SÁNCHEZ FERMÍN
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



RISF/momv*

INSTITUTO TECNOLÓGICO DE HERMOSILLO
DIVISIÓN DE ESTUDIOS DE POSGRADO



Dedicatoria

A mis padres que son mi orgullo y motivación para lograr ser alguien en la vida, que me han inculcado el valor de conseguir las cosas con trabajo y dedicación, a dar todo de mí en cada actividad que realizo y sobre todo por darme la confianza en las decisiones que tomo en mi vida, gracias infinitas.

A mis hermanos que son mi motivo para dar el ejemplo y que al seguir mis pasos sé que debo dejar lo máximo que esté en mí para ellos.

Agradecimientos

A Dios el permitirme la oportunidad de avanzar un escalón en mi proyecto de vida.

A mi director, M.C. César Enrique Rose Gómez, por ser más que un mentor y brindarme su completa confianza en la realización de este trabajo, además de guiarme con sabiduría y permitirme formar un gran lazo de amistad.

A mi codirector, Dr. Samuel González López, por su valioso tiempo invertido en las revisiones y los consejos brindados para la realización de este trabajo.

A mis compañeros de generación, que fueron parte del día a día en el proceso de posgrado, así como a los compañeros de las demás generaciones con los que tuve la oportunidad de convivir y formar buenos vínculos de amistad.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) por recibirme y permitirme realizar una estancia de investigación, y al Dr. Aurelio López López, Jesús Miguel García Gorrostieta y Mario Ezra Aragón por su valioso tiempo, ayuda entera y compartirme su conocimiento en apoyo a mi formación para el trabajo realizado.

Al Instituto Tecnológico de Hermosillo, mi alma mater, por su apoyo en la formación recibida y al núcleo académico por su labor de enseñanza para nosotros.

A Jesús Enrique Villalobos Pereda, por su invaluable labor al buscar la inclusión de la comunidad de Sordos y el apoyo constante en el trabajo de traducciones.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico en la beca no. 627964.

Todos formaron parte invaluable en este proyecto y estaré siempre agradecido, gracias totales.

Resumen

Actualmente, la comunidad de Sordos se enfrenta a diversas problemáticas de comunicación en su entorno, tanto en lo social, en lo laboral y en el ámbito educativo. Se denota el término 'sordo' a la persona con dificultad para el uso del sentido del oído debido a una pérdida parcial o total de la audición. Existen un gran número de personas que cuentan con problemas auditivos y, sin embargo, hay una gran insuficiencia de intérpretes de lengua de señas mexicana (LSM) certificados al contar con solo 45 en el país.

En este trabajo de tesis se presenta la propuesta, análisis, diseño, integración y pruebas de un traductor de texto en español a texto en glosa de LSM el cual llega a ser una herramienta útil para el apoyo en las traducciones que realiza una plataforma de apoyo educativo.

Se muestra la teoría sobre la forma en que se trabajó el módulo de traducción como es el proceso de lenguaje natural, la gramática de la lengua de señas mexicana, la creación de máquinas de traducción mediante modelos tradicionales y modelos de aprendizaje computacional. Además de la implementación y los experimentos realizados, así como el análisis de cada uno de los resultados obtenidos en la experimentación.

El módulo de traducción obtenido muestra diversos resultados en comparación con los diversos modelos que se implementaron confirmando la viabilidad de la obtención de resultados con los modelos tradicionales y de aprendizaje computacional.

Abstract

Nowadays, the Deaf community faces various communication problems in their environment, both socially, in the workplace and in the educational field. The term “deaf” is denoted to the person with difficulty using sense of hearing due to partial or total loss hearing. There are a large number of people who have hearing problems and, nevertheless, there is a great insufficiency of Mexican Sign Language (LSM) Interpreters certified with only 45 in the country.

This thesis paper presents the proposal, analysis, design, integration and testing of a translator from Spanish to Spanish text in gloss of LSM, which becomes a useful tool for supporting translation by an educational support platform.

The theory is shown on how the translation module was worked on, such as the natural language process, the grammar of the Mexican sign language, the creation of translation machines using traditional models and computational learning models. In addition to the implementation and the experiments performed, as well as the analysis of each of the results obtained in the experimentation.

The translation module obtained shows different results compared to the different models that were implemented confirming the feasibility of obtaining results with traditional and computational learning models.

Contenido

Lista de figuras	ix
Lista de tablas	xi
1. Introducción	1
1.1 Introducción	2
1.2 Antecedentes	3
1.3 Planteamiento del problema	5
1.4 Objetivos	6
1.4.1 Objetivo General	6
1.4.2 Objetivo Especifico	7
1.5 Justificación de la tesis	7
1.6 Alcance y delimitaciones	8
1.7 Metodología	8
1.8 Organización de la tesis	9
2. Marco Teórico	11
2.1 Procesamiento de Lenguaje Natural	12
2.1.1 Divisiones en el lenguaje natural	12
2.1.2 Herramientas disponibles para el procesamiento de lenguaje natural	20
2.2 Traducción de texto	22
2.3 Traducción de texto en el contexto de lengua de señas	26
2.3.1 Estructura de la lengua de señas	28
2.3.2 Morfología de la lengua de señas mexicana	29
2.4 Aprendizaje Profundo	29
2.4.1 Redes neuronales artificiales	32
2.4.2 Redes neuronales feed-forward para el aprendizaje profundo	33
2.4.3 Redes neuronales recurrentes para el aprendizaje profundo	35
2.5 Herramientas para el aprendizaje profundo	37
2.5.1 Python para machine learning	37

2.5.2 TensorFlow	39
2.5.3 Keras	41
2.6 Traductores de texto en español a texto LSM	42
3. Análisis y diseño	47
3.1 Modulo de Procesamiento de Lenguaje Natural	49
3.2 Modulo de Análisis Gramatical para LSM	51
3.3 Modulo de Traducción	54
3.4 CORPUS español – lengua de señas	57
4. Integración de Herramientas Computacionales.....	59
4.1 Entorno de desarrollo	60
4.2 Algoritmo para el procesamiento de lenguaje natural	63
4.3 Algoritmo para la alineación de lemas	66
4.4 Algoritmo para la traducción	67
2.3.1 Modelo usando autómatas	68
2.3.2 Modelo usando redes profundas	69
5. Pruebas y análisis de resultados	73
5.1 Formas de evaluación de las traducciones	74
5.1.1 Métricas	74
5.1.2 Evaluación manual	76
5.1.3 Evaluación automática	77
5.2 Preparación de los datos de prueba	77
5.2.1 Obtención de datos de entrenamiento y prueba	80
5.3 Experimentos	81
5.3.1 Experimento con corpus español-inglés	81
5.3.2 Experimento con corpus español-LSE	85
5.3.3 Experimento con corpus español-LSM	87
5.3.4 Experimentos del traductor	88
5.4 Resultados	89
5.4.1 Resultados con corpus español-inglés	89

5.4.2 Resultados con corpus español-LSE	95
5.4.3 Resultados con corpus español-LSM	101
5.4.4 Resultados del traductor	103
5.4.5 Comparación de los resultados	105
6. Conclusiones y trabajo futuro	107
6.1 Conclusiones	108
6.2 Trabajo futuro	110
Bibliografía	111

Lista de figuras

Figura 2.1 Proceso de PLN	13
Figura 2.2 Proceso de traducción	23
Figura 2.3 Estructura de la lengua de señas mexicana	28
Figura 2.4 Métodos de aprendizaje supervisado y no supervisado [47]	31
Figura 2.5 Estructura de neurona biológica y neurona artificial	33
Figura 2.6 Arquitectura de una red neuronal feed-forward.....	35
Figura 2.7 Arquitectura de una red neuronal recurrente	37
Figura 2.8 Arquitectura de TensorFlow [57]	41
Figura 3.1 Arquitectura del traductor de español a texto LSM.....	48
Figura 3.2 Modulo de PLN	50
Figura 3.3 Modulo de análisis gramatical	52
Figura 3.4 Modulo de traducción	55
Figura 3.5 Modelo de traducción con un autómata	55
Figura 3.6 Modelo de traducción con red profunda	56
Figura 3.7 Modelo de evaluación	57
Figura 3.8 Proceso de desarrollo del corpus LSE [69]	58
Figura 4.1 Anaconda Navigator y el entorno de desarrollo	62
Figura 4.2 Fragmento de código de librerías para procesamiento de lenguaje	64
Figura 4.3 Fragmento de código de lectura del archivo y el analizador	65
Figura 4.4 Visualización del texto lematizado	65
Figura 4.5 Fragmento de código de la limpieza gramatical	67
Figura 4.6 Fragmento de código de la traducción con un autómata	68
Figura 4.7 Fragmento de código de la limpieza de datos	69
Figura 4.8 Fragmento de código de generación de datos	70
Figura 4.9 Modelo codificador-decodificador construido	71
Figura 4.10 Fragmento de código de visualización de oraciones	72
Figura 5.1 Fragmento de código para la separación de los conjuntos de datos ...	80
Figura 5.2 Comparación de perdida en entrenamiento y validación del experimento 1.1	90

Figura 5.3 Comparación de perdida en entrenamiento y validación del experimento 1.2	93
Figura 5.4 Comparación de perdida en entrenamiento y validación del experimento 2.1	96
Figura 5.5 Comparación de perdida en entrenamiento y validación del experimento 2.2	99
Figura 5.6 Comparación de perdida en entrenamiento y validación del experimento 3.1	102
Figura 5.7 Evaluación de las traducciones con las métricas	104

Lista de tablas

Tabla 2.1 Herramientas disponibles para PLN [23]	21
Tabla 2.2 Herramientas en línea para la traducción de texto	24
Tabla 2.3 Librerías usadas para Python	39
Tabla 2.4 Estado del arte referente a la traducción de español a lengua de señas	42
Tabla 3.1 Ejemplos de oraciones del corpus	59
Tabla 4.1 Características del equipo de cómputo del proyecto	62
Tabla 5.1 Información corpus español-ingles	78
Tabla 5.2 Información corpus español-LSE	79
Tabla 5.3 Información corpus español-LSM	80
Tabla 5.4 Visualización de las oraciones	82
Tabla 5.5 Distribución del corpus español-inglés en experimento 1.1	83
Tabla 5.6 Distribución del corpus español-inglés en experimento 1.2	84
Tabla 5.7 Distribución del corpus español-LSE en experimentación	86
Tabla 5.8 Distribución del corpus español-LSM en experimentación	88
Tabla 5.9 Resultados de las últimas tres épocas del experimento 1.1	90
Tabla 5.10 Oraciones predichas del experimento 1.1	91
Tabla 5.11 Resultados de las iteraciones del experimento 1.2	92
Tabla 5.12 Oraciones predichas del experimento 1.2	93
Tabla 5.13 Resultados de las iteraciones del experimento 2.1	95
Tabla 5.14 Oraciones obtenidas en experimento 2.1 del corpus esp-LSE	97
Tabla 5.15 Resultados en las iteraciones del experimento 2.2	98
Tabla 5.16 Oraciones obtenidas en experimento 2.2 del corpus esp-LSE	100
Tabla 5.17 Resultado de las iteraciones del experimento 3.1	101
Tabla 5.18 Oración obtenida en el experimento 3.1 con el corpus LSM	103
Tabla 5.19 Oraciones resultado del autómata y red profunda	103
Tabla 5.20 Comparación de resultados	105

CAPÍTULO 1: Introducción

*“La educación es el arma más poderosa
que puedes usar para cambiar el mundo”.
– Nelson Mandela.*

1.1 Introducción

La discapacidad es una condición de vida de una persona que se adquiere al nacer o en cualquier etapa de la vida, esto es manifestado por las limitaciones en el funcionamiento intelectual, motriz, sensorial (vista y oído), entre otros [1].

La discriminación hacia las personas que presentan alguna discapacidad ha sido generalizada y cada vez es más visible este hecho en la sociedad. Cuando vamos por la calle y vemos a una persona con discapacidad visual podemos darnos cuenta por los elementos que los acompañan, ya sea gafas, bastón o algún otro objeto que se puede identificar en la persona, el problema que se tiene.

En una persona con discapacidad auditiva la situación es diferente, ya que es más difícil poder saber que tiene este problema hasta que se tiene la comunicación con dicha persona, y es entonces cuando se hace visible la discapacidad ya que no presenta ningún elemento que nos haga relacionar que la persona tiene la discapacidad auditiva, y al tratar de entablar una comunicación no se puede por el hecho de que muchos no sabemos cómo es la comunicación con este tipo de personas que padecen esta discapacidad.

La Organización Mundial de la Salud (OMS) informó que más de 360 millones de personas en el mundo padecen algún problema auditivo. Por lo tanto, se tiene como consecuencia una gran cantidad de personas con una limitación en la comunicación con los demás. Adicionalmente, lo anterior tiene efectos perjudiciales en el desarrollo intelectual al necesitar mayor asistencia educativa y esto denota como consecuencia tener altos índices de fracaso escolar, ya que esta asistencia no siempre está disponible para todos los estudiantes con problemas auditivos que lo necesitan [2].

En México según el último censo realizado en el 2010 por el Instituto Nacional de Estadística y Geografía (INEGI) de 2, 437,397 personas encuestadas con algún tipo de discapacidad, 498,640 personas padecen la limitación para oír [3]. En Sonora hay en promedio cerca de 7 mil personas con discapacidad auditiva, de los cuales solamente se tienen a 7 estudiando un nivel universitario, *“hacen falta intérpretes para que la comunidad entienda que el sordo es competente a nivel intelectual”* manifestó Raúl Isaac Acedes Valles en un reportaje que se le realizó [4][5].

Lo anterior dio paso a proponer el presente trabajo de tesis para la realización de un traductor de texto en español a texto en lengua de señas mexicano (LSM) y poder adaptarlo a una plataforma educativa con la finalidad de que las personas con discapacidad auditiva puedan seguir con sus estudios sin la problemática de la comunicación a la que se enfrentan.

1.2 Antecedentes

La lengua de señas mexicana (LSM) es la lengua que utilizan las personas sordas en México, y como toda lengua posee su propio léxico, sintaxis y gramática.

La interpretación de LSM surgió como alternativa y necesidad para la comunicación de personas Sordas y que se tuviera una mejor comunicación con las personas oyentes y de esta forma tener las mismas oportunidades sociales que se presentan para ambas comunidades [6].

Las lenguas de señas no son espejos de las lenguas orales que tenemos, sino que tienen sus propias reglas y al ser comparadas se pueden observar significativamente las diferencias que presentan ya que la lengua de señas está basada en varios elementos como lo son señas articuladas de las manos,

movimientos del cuerpo y gestos de la cara que en conjunto forman un significado [7].

En la actualidad existen diversos trabajos que versan en la traducción de las diferentes lenguas de señas que existen alrededor del mundo. Tal es el caso de un trabajo que se realizó en España en el cual se desarrolló un traductor para la interpretación de información en oficinas de renovación de pasaportes.

Además de los diversos trabajos que se realizan alrededor de este tema, en México se realizó un corpus de traducción de texto LSM a texto en español. La variación de este trabajo es que la traducción se realiza en el sentido inverso al buscado por el presente trabajo de tesis dando como resultado una traducción de LSM a texto en español para las personas oyentes que se quieran comunicar con los Sordos [8].

De igual forma se han desarrollado trabajos en nuestro entorno, dentro de los cuales se encuentran los proyectos [70, 71] del Tecnológico Nacional de México campus Instituto Tecnológico de Hermosillo, en donde se destacan dos de ellos acerca de plataformas que apoyan a las personas con discapacidad auditiva en la visualización de la lengua de señas, la cual se realiza por medio de una tableta.

Uno de los proyectos antes mencionados será el punto de partida para el desarrollo del traductor, ya que formará parte del trabajo como el módulo de traducción, donde tendrá como objetivo la realización de las traducciones dentro de la plataforma desarrollada destacando la búsqueda de las mismas haciendo uso de la tecnología de aprendizaje profundo.

Actualmente existen diversos paradigmas de la Inteligencia Artificial (IA) que se basan en generar aprendizaje, ya sea por la forma de ingreso de información para su aprendizaje por parte de una persona que son denominados supervisados, o que

por sí solos generen el aprendizaje que son denominados no supervisados, y que estos últimos son cada vez más buscado en las aplicaciones de IA, con la finalidad de evitar enfrentarse con infinidad de combinaciones de datos que se pueden formular para dar solución a los problemas que se presentan en el mundo real.

El aprendizaje profundo es un aspecto de la IA que busca imitar el enfoque de aprendizaje de los seres humanos construyendo un conjunto de características por sí mismo sin supervisión, creando así modelos predictivos basados en aprender representaciones de datos e interpretarlas para así formular la solución [9].

1.3 Planteamiento del problema

En México actualmente hay insuficiencia de traductores e intérpretes de lenguas de señas nacionales que brinden atención y apoyo en el sector educativo público, a la población que tiene un problema auditivo y que desean desarrollarse profesionalmente.

Cerca de 45 intérpretes certificados en Lengua de Señas Mexicana (LSM) son los existentes en todo el país. En el estado de Sonora hay más de 7 mil personas con problemas auditivos de los cuales se tienen contabilizados alrededor de 2 mil en Hermosillo, lo que significa que a cada uno de los intérpretes les corresponderían 233 Sordos [5].

Estas cifras son muy preocupantes por el poco personal capacitado que existe, por lo cual se propone este proyecto de investigación para trabajar en el desarrollo de un traductor de texto en español, que es el lenguaje en el que se desarrolla la mayor parte del estudio en nuestro país, a texto LSM usando como enfoque el aprendizaje profundo para el procesamiento de las traducciones.

A partir de la problemática que se expone surgen las preguntas de investigación siguientes:

- ¿Qué elementos debe contener un sistema de traducción de texto en lenguaje español a texto LSM?
- ¿Cuál es la estructura del texto LSM?
- ¿Qué tipo de técnicas de aprendizaje profundo son las más adecuadas para la generación de una máquina de traducción?
- ¿Qué elementos debe contener un traductor de texto en español a texto en lengua de señas usando técnicas de aprendizaje profundo?

1.4 Objetivos

Para el diseño y desarrollo del traductor de texto se plantean los siguientes objetivos:

1.4.1 Objetivo general

Diseñar e integrar un modelo de traducción de texto en español a texto LSM aplicando el enfoque de aprendizaje profundo.

1.4.2 Objetivo específico

- Estudiar y analizar los elementos del corpus de texto LSM.
- Estudiar y analizar los elementos para la traducción de texto español a texto LSM.

- Investigar los elementos y características del aprendizaje profundo para su implementación.
- Analizar e implementar la aplicación del aprendizaje profundo en el traductor.
- Implementar y adaptar el uso de un corpus de texto LSM dentro de la aplicación del aprendizaje profundo.

1.5 Justificación de la tesis

Actualmente se desarrolló una plataforma para el ámbito educativo [71] que tiene como objetivo apoyar en la formación educativa a las personas que tienen discapacidad auditiva, por lo cual se propuso esta investigación para integrar el traductor de textos a la plataforma ya mencionada.

Se busca generar un módulo de traducción, empleando la técnica de aprendizaje profundo con el fin de obtener una herramienta capaz de generar traducciones de manera efectiva, y que pueda ser utilizada dentro de una plataforma para apoyo educativo.

De igual forma, se busca acercar la investigación al uso práctico del aprendizaje profundo con la finalidad de aplicar el uso de esta tecnología en el reconocimiento y procesamiento de la información que se va a generar del texto en español a la traducción de texto LSM.

1.6 Alcance y delimitaciones

La estructura de la lengua de señas está basada en tres elementos que son las estructuras de señas formadas por las manos, movimientos corporales y gestos faciales, que en conjunto forman parte de la lengua de señas. Como tal, el presente trabajo se basará solo en las estructuras textuales que forman la lengua,

considerando como trabajo futuro la implementación de los movimientos corporales y gestos faciales que se derivan de la estructura de la lengua de señas.

Por otro lado se cuenta únicamente con un corpus limitado a 229 oraciones de lengua de señas, con el que se trabajara para uso del modelo que se busca generar. Para la plataforma se tiene como apoyo el trabajo de tesis [71] realizado por el Ing. Ernesto Darío Barraza Granillo, además de contar con la colaboración de especialistas en procesamiento de lenguaje natural.

1.7 Metodología

El proceso en el que se desarrolló el presente trabajo de tesis se compone de la siguiente metodología que consta de tres etapas; la primera fue la investigación documental para establecer el estado del arte y el marco referencia, posteriormente se realizó un análisis y diseño del algoritmo y por último la implementación en la que se incluyen pruebas y funcionalidad. Cada una de las etapas mencionadas anteriormente se explicará a continuación:

1. La primera etapa consiste en el estudio de los conceptos teóricos para obtener bases sólidas para llevar acabo el desarrollo de la presente tesis. Los temas principales son: análisis de la lengua de señas mexicana, procesamiento de lenguaje natural, técnicas de aprendizaje profundo y temas en relación con la traducción de texto.
2. En la segunda etapa, se realizó un análisis basado en la información recopilada en el estado del arte donde se definió la arquitectura a generar y se diseñó un modelo de traducción para el presente trabajo.

3. La tercera etapa dio paso a la integración del modelo del traductor, generando cada uno de los módulos que conforman la arquitectura y poder tener un módulo de traducción de texto.
4. Una cuarta y última etapa consiste en la ejecución de pruebas experimentales y análisis de los resultados, donde se pudieron evaluar los modelos de traducción generados en el módulo y de esta forma determinar qué tan efectivos fueron los resultados en la integración de los modelos de traducción seleccionados para el módulo de traducción.

1.8 Organización de la tesis

La organización del presente trabajo de tesis es la siguiente:

El capítulo 1 presenta la problemática con la que se va a trabajar, el objetivo que se desea alcanzar, la justificación del por qué se desea realizar este trabajo, los alcances y delimitaciones que se presentan, la metodología y organización presente de la tesis.

El capítulo 2 en donde se presentan los temas analizados en referencia al marco teórico que se requiere para dar fundamento a la problemática del presente proyecto de tesis.

El capítulo 3 se basa en el desarrollo del traductor con base en el análisis e información que fueron recopilados del marco teórico.

El capítulo 4 presenta la integración del traductor obteniendo como resultado una herramienta que proporciona traducciones de un texto origen a otro texto destino.

El capítulo 5 muestra las pruebas de funcionalidad y los resultados que se obtuvieron de la integración realizada.

Por último, en el capítulo 6 se presentan las conclusiones obtenidas del presente trabajo de tesis, las aportaciones que se pueden generar y el trabajo que se puede realizar en el futuro como aporte de esta investigación.

CAPÍTULO 2: Marco Teórico

“El conocimiento es la mejor inversión que se puede hacer”.

– Abraham Lincoln.

2.1 Procesamiento de lenguaje natural

El vasto campo que se obtiene a través de la interacción e intersección entre la lingüística y la computación es estructurada en varias ciencias, entre las que destaca la lingüística computacional basada en la construcción de modelos de lenguaje que sean entendibles para las computadoras y, por otro lado se encuentra

el procesamiento automático de lenguaje natural (PLN), que se encarga de los aspectos técnicos, matemáticos y algorítmicos implementados en grandes volúmenes de texto, y permite extraer la información útil o inclusive realizar la transformación de esta información, se pueda obtener una estructura formal del texto con la que se pueda lograr el entendimiento e interpretación del lenguaje humano por parte de las computadoras [10].

Hasta hace poco eran los humanos los que intentaban entender los lenguajes informáticos a través de secuencias de comandos y lenguajes de programación, pero en poco tiempo atrás un nuevo fenómeno está ganando fuerza, las máquinas aprenden y entienden el lenguaje de los seres humanos.

2.1.1 Divisiones en el lenguaje natural

Existe la posibilidad de desarrollar un modelo de lenguaje completo, sin embargo, es preferible realizar la división del objeto en partes y de esta manera construir modelos que sean pequeños y a su vez simples de partes del lenguaje.

El entendimiento del lenguaje natural es una tarea compleja que involucra diversos niveles de procesamiento y una variedad de subtarear. En el artículo descrito en [10] se menciona la división para el campo del procesamiento de lenguaje natural de la siguiente forma:

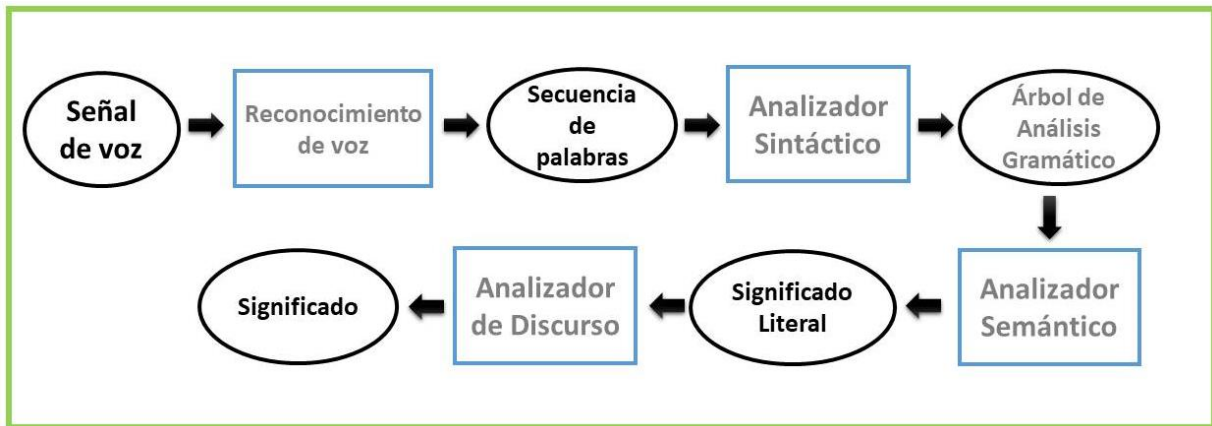


Figura 2.1. Proceso de PLN [28].

Como se muestra en la figura 2.1, se realiza el reconocimiento de voz y análisis de lenguaje hablado por medio de la entrada de una señal de voz, llegando a un módulo de reconocimiento de voz y obteniendo una secuencia de palabras.

De esta forma se obtiene la entrada para el análisis sintáctico, quien al realizar el proceso de agrupación de palabras en componentes para determinar la estructura gramatical, genera una comparación para el árbol de análisis gramático que a su vez será la entrada al análisis semántico.

Se realiza el proceso de desambiguación palabra-sentido por parte del analizador semántico y lo proporciona al análisis del discurso y extracción de información con el que, por medio del significado, se llega a la traducción automática exacta.

Con respecto al tema de la lingüística computacional está presente la necesidad de desarrollar varios recursos lingüísticos, entre los que se encuentran los corpus con diferentes tipos de análisis para la traducción con el procesamiento de lenguaje como lo son los fonéticos, morfológicos, sintácticos, semánticos, de sentidos de

palabras, entre otros, los cuales son definidos a continuación, haciendo énfasis en los componentes de un sistema general de comprensión lingüística como el mostrado en la figura 2.1.

- Fonética

Es la parte de la lingüística dedicada a la exploración de las características del sonido, al ser un elemento de la lingüística en su mayoría físico, obliga que su posición dentro de la lingüística sea independiente en comparación con los demás recursos lingüísticos.

Existen problemas fonéticos computacionales que se relacionan con el desarrollo de sistemas de reconocimiento de voz y síntesis de habla. Si bien, hay sistemas de reconocimiento de voz, como las que se generan al reconocer las palabras con ayuda del micrófono de la computadora o el celular, en el que se pueden obtener porcentajes de reconocimiento de palabras considerablemente bajos, sin embargo, dentro de los sistemas de síntesis de habla hay mayor éxito en este aspecto.

Podemos encontrar herramientas que realizan el reconocimiento de voz como es el caso de Microsoft Cognitive Services que cuenta con una API denominada *Bing Speech API*, cuya aplicación tiene la función de reconocimiento de audio en tiempo real y posteriormente realiza la traducción a texto, además de permitir el reconocimiento de audios dentro de archivos [11].

- Morfología

Tanto para la preparación de los corpus como para la preparación de análisis lingüísticos, una etapa indispensable es la detección de características morfológicas

de las palabras, es decir, el análisis morfológico o léxico. Puede ser sustituido o simulado por diversas técnicas, sin embargo, la calidad del análisis y de los recursos son mejorados considerablemente con un análisis morfológico exacto [28].

La morfología estudia la correcta estructura de las palabras y su relación con las categorías gramaticales de la lengua. Su objetivo es llevar a cabo la clasificación de la palabra que conforma una oración de forma específica.

Existen criterios de clasificación según corresponda la palabra y se pueden observar de la siguiente manera: mediante morfemas diminutivos, como lo son cosita, pequeñita, monedita cuando se habla con el morfema “-ita”; mediante el morfema aumentativo “-ote” como fuertote, grandote, malote; en donde la clasificación es realizada mediante el morfema femenino con terminación “-a” como silla, mesa, casa, cama, rosa; también entra el criterio por los tiempos donde se usa el criterio morfológico por la terminación “-e” u “-o” para dar noción al tiempo pasado como lo es lavé, disfruté, cociné o luché, comió, jugó, entre otros ejemplos [28].

Para el español existen varios sistemas que generan el análisis morfológico, por ejemplo: MACO, FreeLing, FLANOM, o incluso el analizador morfológico integrado en MS Word, algunos de ellos explicados más adelante. Si se desea realizar un módulo independiente para el funcionamiento del sistema que se está desarrollando, se recomienda hacer uso de herramientas que estén disponibles para aplicarse como módulos externos (la herramienta FreeLing es de disponibilidad libre y se puede adaptar de forma modular), ya que al hacer uso de analizadores morfológicos web (como el caso de MACO), se vuelve dependiente el

funcionamiento a factores que repercuten de forma considerable si no se cuenta con ellos [28].

Al realizar la implementación de los analizadores morfológicos automáticos debemos distinguir de manera importante: el modelo de análisis (el procedimiento de análisis), el modelo de gramática que se emplea en el analizador (las clases gramaticales de palabras) y la implementación computacional (el formalismo usado).

- Sintaxis

Este nivel es uno de los cuales, durante mucho tiempo y aún vigente, la lingüística se ha dedicado a prestar mayor atención en referencia al tratamiento de lenguaje natural. A grandes rasgos, la sintaxis se basa en el análisis de las relaciones entre las palabras dentro de la frase y de esta manera saber si la estructura gramatical de la oración es correcta. Debe contener métodos para el análisis y síntesis automática, que le permita construir la estructura de la frase, o bien, generar la frase basándose en el mismo orden [28].

Pese a la gran importancia que representa el analizador sintáctico o también denominado “parser”, aún es un problema abierto, especialmente para los idiomas que no llevan un orden fijo en las palabras que usan en una oración, como lo es el español. Las razones para usar el analizador sintáctico son justificadas de la siguiente manera:

- El proceso semántico funciona sobre los componentes de la oración. Cuando no existe un paso de análisis sintáctico, el sistema semántico debe identificar sus

propios componentes. Por otra parte, si se realiza un análisis sintáctico se restringe el número de componentes a considerar por el semántico y no obstante, computacionalmente hablando, la existencia de un análisis sintáctico conlleva un considerable ahorro de recursos y reduce la complejidad del sistema [12].

- Aunque frecuentemente se puede obtener el significado de una oración sin hacer uso de hechos gramaticales, pueden presentarse casos donde no se pueda realizar este proceso [13].

Existen algunas herramientas disponibles que pueden utilizarse para la realización del proceso sintáctico entre los que se muestran 'BISON' [14] y 'COCO/R' [15], escritos en lenguaje C, como analizadores sintácticos que en su mayor uso se encuentra la construcción de compiladores.

- Semántica

Hay que saber el sentido de todas las palabras e interpretar las relaciones sintácticas, lo que nos lleva al propósito de la semántica el cual es entender la frase y detectar el sentido de cada elemento dentro de la oración, proporcionando el significado de las oraciones una vez eliminadas las ambigüedades morfo-sintácticas [28].

Dado el anterior concepto, podemos observar un ejemplo sobre la aplicación del análisis semántico para conseguir la desambiguación de sentidos de palabras. Si tomamos la frase 'un gato' podemos determinar dos tipos de cosas, un felino y una herramienta mecánica. Para saber el sentido del contexto de la palabra se analizan las demás palabras que se presentan en determinado contexto, por ejemplo, si nos

dicen 'El gato está maullando y caminando en la barda', podemos deducir que el maullando y caminando corresponden a un ser vivo felino, en cambio, si tenemos la oración 'El mecánico uso el gato para subir la camioneta', haciendo uso de las palabras mecánico y camioneta nos da como referencia a una herramienta para apoyo en la mecánica automotriz.

Un ejemplo de uso es el motor de búsqueda de Google, el cual hace uso de la semántica para realizar "búsquedas más inteligentes" para los usuarios, analizando el contexto general de las palabras buscadas y agregando algunas características te muestra resultados relevantes a lo deseado. Esto permite diferenciar entre dos cosas, la intención y el contexto, donde el primero es lo que se está buscando, mientras que el segundo es lo que se quiere buscar [16].

Algunas de las herramientas que mejoran la búsqueda semántica son '*Answer the public*' [17] y '*Keyword Tool*' [18], las cuales proporcionan ayuda como asignación de sugerencias de palabras que proporcionan información sobre las motivaciones detrás de la búsqueda de las personas y permiten recopilar palabras clave para realizar las autosugerencias a futuro. Otra es '*Spanish WordNet*' [19] la cual es una base de datos léxica que organiza sustantivos, verbos y adjetivos en conjuntos de sinónimos e indica las relaciones semánticas entre ellos.

- *Pragmática*

La pragmática es una rama de la lingüística que maneja el lenguaje según su contexto e incorpora el análisis del contexto de uso a la interpretación final en el estudio de su sentido, tanto el comunicado por el hablante (o escritor) y el interpretado por el oyente (o lector).

Ha sido un elemento cuestionado sobre si forma parte del grupo que conforma la lingüística o bien si forma parte de un área transversal a todos los componentes que conforman la lingüística situado entre las ciencias cognitivas, filosofía y otras disciplinas. Eso nos lleva a [20], que ha conceptualizado de la siguiente manera la pragmática: “*Con el tiempo la pragmática ha dejado de ser un componente más de la teoría lingüística para convertirse en perspectiva sobre el lenguaje, en un marco teórico para interpretar la comunicación*” [21].

Lo que es sabido, dentro de la pragmática confluyen áreas disciplinares muy diversas y cada una de ellas ha contribuido de un modo u otro a su desarrollo. Por ello resulta la importancia de incluir esta disciplina como un componente fundamental para la perspectiva de comunicación.

- Discurso

La comprensión del lenguaje es un proceso donde un sujeto interpreta una frase u oración, ya sea al leerla o escucharla. El análisis del discurso implica determinar el contexto que influye dentro de la interpretación de esa frase u oración. Normalmente tratamos con varias oraciones con cierta relación entre sí, las cuales al hilarse forman una nueva entidad llamada *discurso*. Dentro de esta tarea, se encuentra un objetivo muy importante a solucionar, la resolución de correferencia [22].

Podemos ver un ejemplo, en el discurso “*Ayer compré un automóvil nuevo. Su color y forma es increíble.*” Se observa la segunda oración en el artículo ‘su’ haciendo referencia a ‘el automóvil’, que es una relación de correferencia y aquí es donde la computadora tiene el trabajo de analizar correctamente para generar las representaciones semánticas.

2.1.2 Herramientas disponibles para el procesamiento de lenguaje natural

Existen diversas herramientas que se encargan de realizar la segmentación y la tokenización de las palabras, algunos de los cuales están integrados en herramientas de gran alcance. En la tabla 2.1 se presenta un resumen de uso general de algunas de las herramientas existentes.

Tabla 2.1. Herramientas disponibles para PLN [23].

Herramienta	Lenguaje	Descripción
NLTK [24]	Python	Natural Language Toolkit (NLTK) es una plataforma de código abierto para realizar tareas de PLN, incluyendo tokenización, derivación, etiquetado gramatical, análisis y razonamiento semántico. Proporciona interfaces para muchos corpus y léxicos que son útiles para la minería de opinión y el análisis de sentimientos. http://www.nltk.org/

OpenNLP [25]	JAVA	<p>El Apache OpenNLP es una librería en Java para el procesamiento de textos en lenguaje natural, que soporta tareas comunes incluyendo tokenización, segmentación de frases, etiquetado gramatical, reconocimiento de entidades nombradas, análisis y resolución de correferencia.</p> <p>https://opennlp.apache.org</p>
CoreNLP [26]	JAVA	<p>Stanford CoreNLP es un framework que soporta no solo la tarea básica de PLN, como el etiquetado gramatical, la redefinición de entidades nombradas, el parseo, la resolución de correferencia, sino también el análisis avanzado de sentimientos.</p> <p>http://stanfordnlp.github.io/CoreNLP/</p>
FreeLing [27]	C++/ MS-Visual/ Java / Python	<p>FreeLing es una librería de código abierto para el procesamiento multilingüe, que proporciona una potente herramienta para desarrollar aplicaciones de PLN en diversos lenguajes. Tiene una amplia gama de tareas como tokenización, diccionario morfológico, detección multipalabra, etiquetado gramatical, resolución de correferencia entre otras.</p> <p>http://nlp.lsi.upc.edu/freeling</p>

2.2 Traducción de texto

Durante siglos, la traducción ha sido caracterizada por ser una mera reproducción lingüística de un texto de partida escrito en lengua extranjera a un texto de llegada escrito en lengua materna. En el ámbito de la computación, el concepto de traducción hace referencia al proceso que efectúa una aplicación de computadora al realizar la conversión de un texto en lenguaje natural escrito en cualquier lenguaje

(fuente) a otro lenguaje (destino). El objetivo principal de las técnicas de traducción es “capturar” el significado original del texto y transmitirlo como un texto coherente en otro idioma [28].

Hay cuestiones justificables sobre la realización de la traducción, como expresa [29], “*se traduce por que las lenguas y culturas son diferentes*” por lo que denomina la razón de ser de la traducción, la diferencia lingüística y cultural, “*se traduce para comunicar*”, debido a que se tiene diferencias lingüísticas y culturales alrededor del mundo, la traducción tiene la finalidad comunicativa, “*se traduce para alguien que no conoce la lengua*” ya que el traductor no traduce para sí mismo, se busca un destino que necesite de él como mediador lingüístico y cultural.

Al revisar las principales propuestas en torno al análisis del proceso traductor, se llega a coincidir en que la traducción es un caso especial de comportamiento cognitivo, formado por el procesamiento humano de información. De esto se llega a distinguir los procesos que lo conforman: *análisis*, *representación semántica* y *síntesis*, además de incluir una fase más de verificación denominada *revisión*, esquematizado en la figura 2.2.

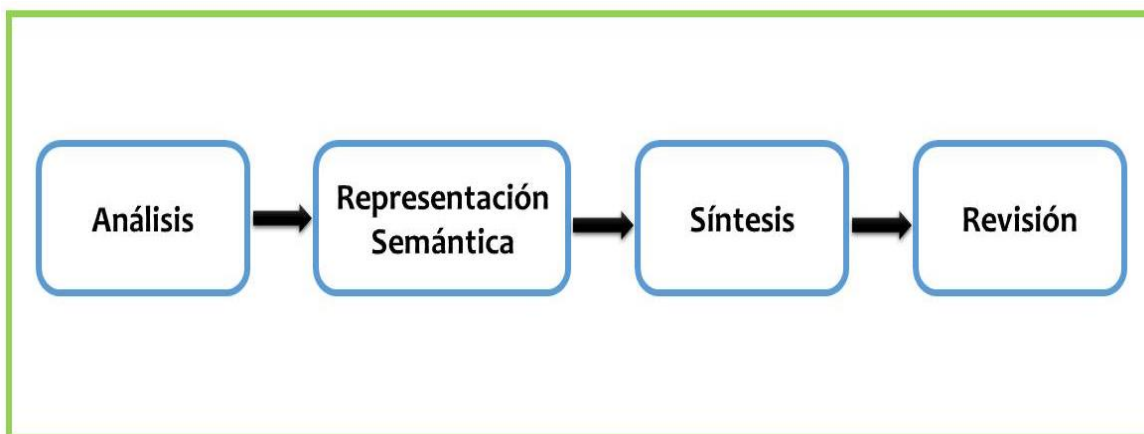


Figura 2.2. Proceso de traducción.

Dentro del proceso de análisis, se interpreta el sentido que tiene el texto original estableciendo los sentidos ya sea de la palabra, frase o más oraciones. Siguiendo con la representación semántica, el traductor conforma en su cabeza un mapa semántico con los significados libres de la lengua de partida como la de llegada. En la fase de síntesis se lleva a cabo el proceso de síntesis y semántico, obteniendo la llegada a la lengua a traducir. Cabe señalar que en esta fase también se verifican rasgos de género textual, reglas gramaticales, el contexto, entre algunas por mencionar. Y de esta manera se finaliza con la revisión. Es importante destacar que el proceso que se explica en la figura 2.2 no necesariamente lleva ese orden, el proceso puede cambiar y modificarse según sea conveniente, sin embargo, seguirá siendo necesario que los cuatro procesos se desarrollen para poder lograr la traducción [30].

La traducción automática no es tan simple como parece a simple vista. No solo se trata de importar grandes diccionarios al motor de traducción automática, sino también hay que enseñarle al sistema a reconocer las colocaciones de las frases y hay que almacenar las reglas como un programa para que se realice la traducción conforme a ese orden.

Actualmente podemos darnos cuenta de las ventajas que existen entre los traductores de hoy en día sobre los traductores del pasado, ya que existen recursos que ayudan en el proceso de traducción como lo es el internet. En la tabla 2.2 se presentan algunas de las herramientas en línea existentes como apoyo para realizar

el proceso de traducción de un texto a otro, con la finalidad de mostrar la forma en la que se exponen las diferentes y variadas traducciones que tienen a su disponibilidad.

Tabla 2.2. Herramientas en línea para la traducción de texto.

Herramienta	Nombre	Descripción
1. Diccionarios de lengua general:	DRAE (Diccionario de la Real Academia Española) [31]	Parte de las publicaciones que ha tenido la real academia española y la asociación de academias de la lengua española, es su diccionario (ahora DLE) que ofrece pronunciaciones, acepciones, sinónimos y ejemplos que favorecen la claridad de las palabras entrantes del léxico general utilizado en España y países hispanos. Existe una versión móvil y una descarga para consultarlo sin conexión a internet. http://dle.rae.es/
	Cobuild [32]	El acrónimo se las siglas para la base de datos de idiomas internacionales de Collins Birmingham University. Es una base de datos analítica de inglés con 4.5 mil millones de palabras. Contiene información escrita de sitios web, periódicos, revistas y libros publicados en todo el mundo. https://collins.co.uk/
	Wordreference [33]	Es una colección de diccionarios en línea que incluye más de 15 idiomas. Algunas combinaciones de idiomas son muy robustas (español e inglés) mientras que otras combinaciones todavía se encuentran en trabajo. http://www.wordreference.com/
	Google Translate [34]	La compañía del buscador más famoso en la actualidad, Google, ha creado una potente herramienta de traducción multilinguaje con la cual se pueden realizar múltiples tipos de traducciones desde palabras escritas, imágenes, texto en aplicaciones, traducción por voz, traducción de una conversación bilingüe, traducir mensajes de texto, páginas web o documentos,

2. Traductores Automáticos:		<p>hasta escrituras a mano o en un teclado virtual. Ofrece un sitio web con una interfaz para traducción, aplicación móvil para iOS y Android, y una API que ayuda a los desarrolladores a construir aplicaciones de software.</p> <p>https://translate.google.com/</p>
	Bing Translate [35]	<p>Es un servicio proporcionado por Microsoft que permite a los usuarios realizar la traducción de textos o páginas web completas en diferentes idiomas. La traducción de textos que presenta es realizada por la traducción automática estadística basada en sintaxis de la tecnología de Microsoft. Tiene la opción de 62 lenguajes disponibles y cuenta con una API cliente de traducción de texto por parte de Microsoft Cognitive Services.</p> <p>https://www.bing.com/translator</p>
	Prompt [36]	<p>Dentro de las máquinas de traducción se encuentra PROMT, una solución lingüística de TI la cual ofrece un sistema de traducción con más de 20 idiomas, encontrando entre los principales el inglés, ruso, alemán, francés, español, italiano, portugués, chino tradicional, ucraniano, turco, japonés y polaco entre otros, ofreciendo sus servicios de manera web, aplicación de escritorio y móvil. Cuenta con un sitio web, el cual provee un servicio de traducción para 60 lenguajes.</p> <p>https://m.online-translator.com/</p>
3. Verificadores Ortográficos:	Microsoft Word [37]	<p>Existente en el procesador de textos Word, el cual con ayuda de la herramienta <i>ortografía</i>, permite al usuario identificar y corregir las palabras mal escritas, así como oraciones que tienen errores gramaticales por medio de un diccionario interno del mismo procesador de textos. Por otro lado, se tiene una herramienta <i>gramática</i> la cual realiza una revisión del cómo se encuentra construido y que estilo tiene la oración del documento dando sugerencia de una oración correcta.</p> <p>https://products.office.com/es-mx/home</p>
	OpenOffice [38]	<p>Dentro de las herramientas existentes de código libre, se encuentra este procesador de textos, que utiliza un motor de corrección ortográfica denominado Hunspell</p>

		(corrector ortográfico y analizador morfológico utilizado en LibreOffice, Thunderbird, Firefox, Opera, Google Chrome entre otros) y con ayuda de complementos externos realiza la revisión de texto de diferentes formas. El corrector que utiliza OpenOffice es basado en reglas el cual permite identificar los problemas gramaticales existentes en el texto. https://www.openoffice.org/
--	--	---

2.3 Traducción de texto en el contexto de lengua de señas

La lengua de señas es el lenguaje natural para las personas sordas y como todas las lenguas emerge de la necesidad de establecer comunicación entre las personas [39]. La lengua de señas no es universal, por lo tanto se tienen diferentes formas de dirigir el trabajo de los traductores así como los lenguajes hablados.

Al hacer referencia a la traducción de lenguas verbales que cuentan con sistema de escritura, se determina el trabajo que representa realizar la construcción de la escritura de lengua a lengua, y es aún más trabajo traducir el sistema de contenidos entre lenguas con diferente sustancia de la expresión como lo son las lenguas de señas [40].

Uno de los ejemplos de aplicaciones que se basan en la traducción de la lengua de señas es *Signslator*, desarrollado por la Asociación para la Normalización del Lenguaje de Señas (ANICOLS) en colaboración con la agencia de comunicación TBWA en España, el cual se basa en la traducción de lengua de señas española. Está basada en un servicio gratuito el cual se encuentra disponible tanto en web como en aplicación para dispositivos móviles, y con los más de 12,000 gestos

grabados que contiene la aplicación, permite mostrar el trabajo como traductor automático [41].

Otro ejemplo de un sistema para la conversión de texto a lengua de señas es TextoSIGN [42], el cual permite realizar la conversión de texto a Lengua de Señas Española. Fue desarrollada por el Instituto Tecnológico de Castilla y León y XUL Social Media Company en el 2011. Contiene 350 signos que interpreta en tiempo real, además de que si un signo no es encontrado, el sistema permite deletrear. Para conocer la traducción a lengua de señas española (LSE) de una palabra, se ocupa introducir en castellano en el cuadro de búsqueda y de esta forma se podrá ver un video con el avatar animado signando la palabra. Una de las ventajas considerables para el uso de este software es que las animaciones y las transiciones se proporcionan con gran realismo.

2.3.1 Estructura de la lengua de señas

Dentro de la lengua de señas se ha tenido poca investigación referente al orden de las palabras. No obstante que ya se comenzaba el estudio desde la década de los setentas, aún no se han dado conclusiones concisas sobre el mecanismo que subyace a la organización de las palabras de las lenguas viso-gestuales.

El autor [43] hace referencia, de manera general, a una lengua con orden Sujeto-Verbo-Objeto (SVO), a pesar de que otros autores como [44] reconocen el orden Sujeto-Objeto-Verbo (SOV), sin embargo, la afirmación de un orden SVO se basa

en gran parte al análisis sistemático de una sola lengua de señas, la *Lengua de Señas Americana* (ASL).



Figura 2.3. Estructura de la lengua de señas mexicana.

Haciendo enfoque en la LSM, y tomando en cuenta que el verbo puede influir en el orden que presenta, de manera general se puede decir que el orden que se observa de constituyentes en las construcciones gramaticales es SVO, mostrado en la figura 2.3, pero también se pueden tener las diferentes variaciones OSV, VOS, VSO, OVS y SOV, según el verbo utilizado así como situaciones semánticas y pragmáticas [45].

2.3.2 Morfología de la lengua de señas mexicana

De manera general la morfología se define como una disciplina lingüística que estudia la estructura interna formal de la palabra y la relación que tiene con el significado. Esto conlleva a que los investigadores de las lenguas de señas se den a la tarea de definir que es una palabra dentro de este tipo de lenguas.

Observando cómo está compuesto y estructurado una lengua de señas, es necesario verificar los criterios semánticos y estructurales que se presentan en el reconocimiento de una palabra, además de que es necesario considerar la noción

del seÑante/hablante en el reconocimiento de qué es una palabra en la lengua que él reconoce [7].

En la morfología de las lenguas de seÑas, en particular de la LSM, no solo existe una concatenación de los elementos, sino que también hay una superposición de ellos, ya que se caracteriza la espacialidad, simultaneidad y temporalidad, además de la secuencia de sus distintos componentes [7].

2.4 Aprendizaje Profundo

El aprendizaje profundo, del inglés *deep learning*, tiene una larga historia y muchas aspiraciones. En la actualidad, compone una de las áreas de investigación más interesantes que se encuentra obteniendo resultados muy buenos sobre el aprendizaje automático. Existen diferentes términos que hablan sobre la correcta definición del aprendizaje profundo, sin embargo, se acepta que se trata de modelos de aprendizaje automático basados en redes neuronales con una gran cantidad de capas no lineales, que tienen la tarea de ir aprendiendo a un nivel de abstracción más alto que sus capas anteriores, llegando a procesar estructuras complejas como semántica, figuras, procesamiento de seÑales, etc. [46].

Históricamente, este concepto fue originario de las investigaciones que se realizaron sobre las redes neuronales artificiales y un ejemplo de modelo de arquitectura profunda son los perceptrones multicapa que a menudo son conocidas como redes neuronales profundas.

Como se puede observar en la figura 2.4, existen dos métodos de aprendizaje, el supervisado y el no supervisado. El aprendizaje supervisado está basado en aprendizaje que es usado para realizar la correspondencia en los conjuntos de entradas y las salidas deseadas, obteniendo una minimización en base a una función de error que mida la diferencia existente entre las salidas y las obtenidas por la red, como se puede encontrar en el aprendizaje automático. Por otro lado, el aprendizaje no supervisado trata de representar los datos de entrada de modo a que estos reflejen la estructura estadística que los define, tratando de entender patrones o características que sean significativos en los datos de entrada. Estos no tendrán salida con la cual comparar el rendimiento del modelo, así que solo tendrá los datos de entrada como información, y de esta forma se encuentra el aprendizaje profundo.

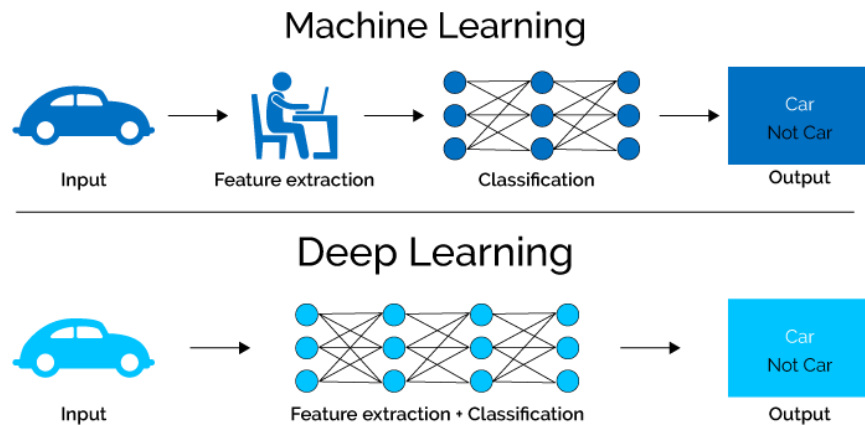


Figura 2.4. Métodos de aprendizaje supervisado y no supervisado [47].

Entre los modelos de aprendizaje profundo y aplicación en procesamiento de texto destaca el uso del perceptrón multicapa o redes feed-forward y redes recurrentes, de las cuales se describen más adelante. Al trabajar con el procesamiento de texto,

debemos considerar ciertos aspectos que conforman el corpus para poder elegir el modelo de aprendizaje a usar, tomando en consideración el tamaño del vocabulario que se va a trabajar, el número o distribución de muestras, obtener la topología de red a usar, el número de capas y neuronas por capas, entre otros aspectos.

Como se analizó anteriormente, el aprendizaje profundo contiene un campo bastante amplio de técnicas de aprendizaje de máquinas y arquitecturas con su etiqueta que es el usar muchas capas de procesamiento no lineales para obtener la información. Todo dependerá de cómo esté diseñada la arquitectura y las técnicas para su uso [48].

2.4.1 Redes neuronales artificiales

En la actualidad, existen desarrollos de científicos que se enfocan en el estudio de las capacidades humanas, tomándolas como modelos para poder realizar implementaciones en las máquinas. Así es el punto de partida de la inteligencia artificial, con el intento de simular la inteligencia humana por medio de máquinas, lo que ha permitido abrir desarrollos en los últimos años en campos como el procesamiento de información expresada con lenguaje humano, la visión artificial, entre otros.

El propósito de las redes neuronales se basa en encontrar la forma de emular la capacidad de aprendizaje de los humanos y poder realizar asociaciones a hechos. Según el concepto de [49] una red neuronal es: “*un sistema para el tratamiento de*

información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso: la neurona”.

Así mismo, la forma de trabajo de una neurona biológica se presenta de la siguiente manera: la neurona es excitada o estimulada por medio de sus *entradas*, y al alcanzar un umbral, provoca la activación de la neurona pasando una señal hacia el *axón*. Por otro lado, el modelo de una neurona artificial se encuentra basado en el de una neurona biológica. La neurona artificial está conformada por las siguientes características: la neurona recibe las entradas (x) que son capturadas por las dendritas, a cada entrada se le agrega un peso (w) y se emplea por la intensidad de la sinapsis o valor de la información para conectar dos neuronas, además se agrega una entrada extra denominada bias (b) que tiene el objetivo de lograr una convergencia más rápida de la red. La llegada que tiene la entrada y el peso es hacia una función umbral, donde la neurona debe de ser sobrepasada para poder activarse. En la figura 2.5 podemos observar la comparación de la neurona biológica con la representación de una neurona artificial.

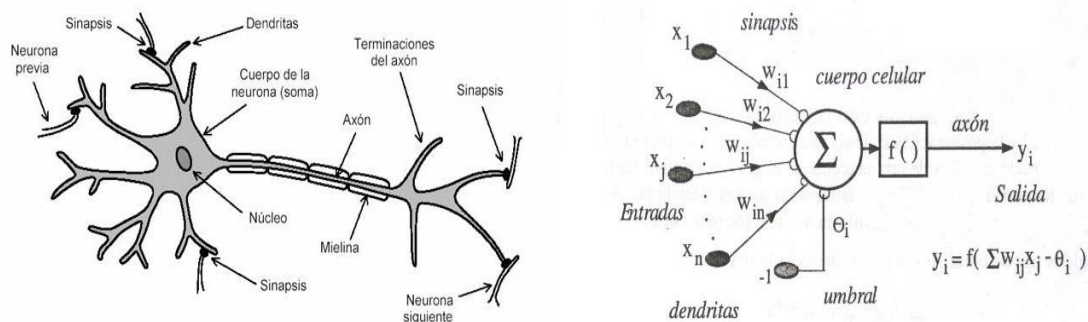


Figura 2.5. Estructura de neurona biológica y neurona artificial.

Una de las características que se encuentran al simular el cerebro humano es *el aprendizaje*, el cual se emplea para resolver problemas que inicialmente no pueden solucionarse, al adquirir mayor información acerca del problema se puede mejorar su funcionamiento y así lograr obtener un mejor resultado al problema.

2.4.2 Redes neuronales feed-forward para el aprendizaje profundo

Las redes feed-forward profundas, también llamadas a menudo *redes neuronales feed-forward* o *perceptrón multicapa* (MLPs), son los modelos de aprendizaje profundo por excelencia. La meta de una red feed-forward es aproximar alguna función f^* [9].

Los modelos feed-forward son denominados de esta manera ya que la información fluye a través de la función que es evaluada desde x , a través de los cálculos intermedios utilizados para definir f , y finalmente llegar a una salida y . En estas redes no hay conexiones de retroalimentación en las que las salidas del modelo se retroalimentan entre sí, ya que cuando las redes neuronales feed-forward son ampliadas para incluir conexiones de retroalimentación son denominadas redes neuronales recurrentes, las cuales son definidas en el apartado posterior.

El concepto de redes neuronales feed-forward está asociado a la representación de la composición de juntar muchas funciones diferentes. Por ejemplo, al tener tres funciones $f_{(1)}$, $f_{(2)}$ y $f_{(3)}$ conectadas en una cadena para formar $f(x) = f_{(3)}(f_{(2)}(f_{(1)}(x)))$, que son las estructuras más comunes de encontrar en redes neuronales. En este caso, es llamada la función $f_{(1)}$ asociándola a la primera capa de la red, la

función $f^{(2)}$ a la segunda capa, y así sucesivamente según se tenga el número de funciones, queda determinando que la longitud de la cadena dará la profundidad del modelo de esta manera se presenta el “aprendizaje profundo” [9].

En la figura 2.6 podemos observar la arquitectura de una red feed-forward, que está basada en una capa de entrada, también denominada capa 0, en la cual no existe procesamiento alguno, su función es realizar la distribución de la entrada hacia la próxima capa del vector de entrada ‘x’. Después se encuentra la capa oculta, que se encuentra ubicada entre la capa de entrada y la capa de salida, en donde se encuentran los nodos de red que son no observables. Cada neurona es una función de la suma ponderada de las entradas, si la red contiene una segunda capa oculta, cada neurona recibe como entrada la salida de todas las neuronas de la capa oculta anterior para la función de activación, logrando así la abstracción a detalle que caracteriza este modelo. Por último se encuentra la capa de salida, esta proporciona los resultados de los datos que fueron previamente procesados y entrega el vector de salida ‘y’.

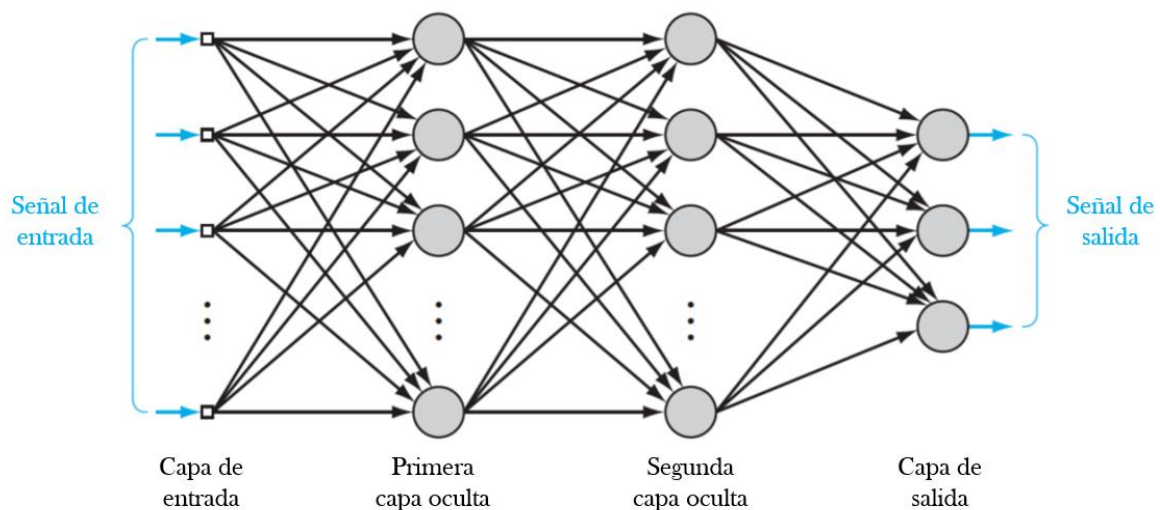


Figura 2.6. Arquitectura de una red neuronal feed-forward.

El método de aprendizaje de las redes neuronales feed-forward es el algoritmo de retropropagación o backpropagation, que se basa en minimizar la función de costo $E(w)$, siguiendo la dirección del gradiente descendiente.

2.4.3 Redes neuronales recurrentes para el aprendizaje profundo

Son redes identificadas debido a que pueden tener conexiones que permiten el regreso desde los nodos de salida con dirección a los nodos de entrada, además de que pueden realizar conexiones arbitrarias a través de sus nodos. Su distinción a otras redes se observa en que las redes recurrentes tienen la capacidad de mantener un estado interno en función de una secuencia de entrada, por lo que pueden modificar su estado interno en medida que se va presentando la información [9].

Este tipo de redes son utilizadas resolviendo problemas en donde la salida no tiene dependencia únicamente de la entrada actual, sino también de otras entradas. Un ejemplo de aplicación es del pronóstico de clima, que en base a los datos de días anteriores, arroja una predicción para el clima de los próximos días.

Su estructura es similar a la red feed-forward, la diferencia es que tienen caminos de retroalimentación entre todos los elementos que la conforman. Como podemos observar en la figura 2.7, al ingresar la señal de entrada llega a la capa de entrada, en donde las neuronas están conectadas a las neuronas posteriores en la siguiente

capa, estas neuronas a su vez conectadas a la capa anterior y a ella misma a través de vectores de pesos variables que son modificados en cada época con la finalidad de alcanzar los parámetros deseados de la operación.

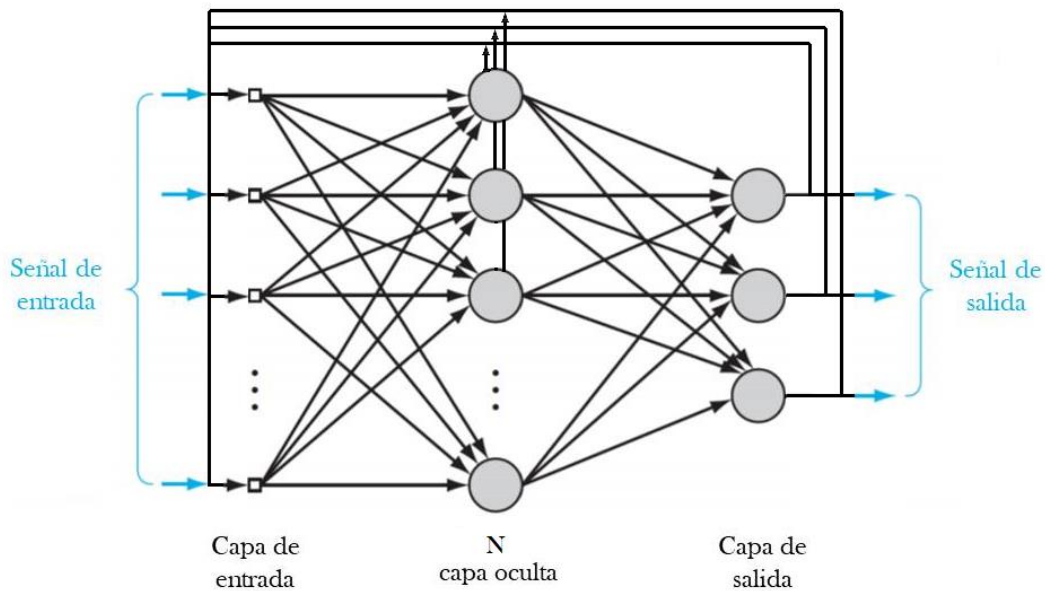


Figura 2.7. Arquitectura de una red neuronal recurrente.

2.5 Herramientas para el aprendizaje profundo

En la actualidad, existen herramientas que nos ayudan a realizar el trabajo de procesamiento de lenguaje natural haciendo uso de redes neuronales profundas.

Los apartados que se presentan a continuación enumeran algunas de las herramientas que pueden ser utilizadas en el área de aprendizaje profundo:

2.5.1 Python para machine learning

Python es uno de los lenguajes de programación más populares en la actualidad, usado para abordar la ciencia de datos. Lo que lo hace una herramienta utilizada es que se puede disfrutar de un gran número de bibliotecas desarrolladas en código abierto.

Aunque el rendimiento de los lenguajes interpretados, como Python, para tareas que requieren mucha complejidad es inferior a los lenguajes de programación de menor nivel, se han desarrollado bibliotecas de extensión que se basan en la capa baja Fortran y C siendo implementaciones para operaciones rápidas y vectorizadas en matrices multidimensionales [50]. Para tareas como el aprendizaje de máquinas, también existen bibliotecas accesibles como '*Scikit-Learn*' la cual es una de las más populares y usadas para realizar este trabajo, además de que es encontrada en código abierto.

Un gestor de paquetes altamente recomendado de Python para la informática científica es '*Anaconda*', que es una distribución gratuita y de uso comercial para el uso de Python que incluye todos los paquetes para trabajar en las áreas que nos permite este lenguaje de programación y cuenta con una interfaz amigable para el usuario.

La tabla 2.3 nos muestra algunas librerías que podemos encontrar para Python.

Tabla 2.3. Librerías usadas para Python.

Nombre	Características	Comando de Instalación
Jupyter [51].	Es una aplicación web que permite crear y compartir documentos. Ofrece funcionalidad de Python en el navegador, lo que permite documentar y reproducir la información de una forma sencilla.	python3 -m pip install --upgrade pip python3 -m pip install jupyter
NumPy [50].	Se trata de una librería de informática científica con soporte para matrices multidimensionales y álgebra lineal, requerida para probabilidades, etiquetado, clustering, clasificación de tareas, entre otras.	git clone https:// github.com / numpy / numpy.git numpy
Matplotlib [52].	Se trata de una librería de graficación 2D para la visualización de datos. También se usa para producir gráficas de líneas y gráficas de barras.	python3 -m pip install --upgrade pip python3 -m pip install matplotlib
Pandas [53].	Se trata de una librería que proporciona análisis de datos, estructuras de datos fáciles de implementar y de alto rendimiento.	conda install pandas pip install pandas
Scikit-Learn [54].	Se trata de una librería con una colección de algoritmos y herramientas para el aprendizaje de máquina, minería de datos y análisis de datos.	conda install scikit-learn pip install -U scikit-learn
SciPy [55].	Se trata de una librería que define algoritmos con los cuales se puede realizar optimización, álgebra lineal, interpolación, procesamiento de señales, integración numérica entre otras.	git clone https:// github.com / scipy / scipy.git scipy

2.5.2 TensorFlow

Es una interfaz escalable y multiplataforma programable para implementar y correr algoritmos de aprendizaje automático, actualmente siendo una de las más populares librerías de aprendizaje profundo disponibles, que pueden implementar redes neuronales más eficientemente.

TensorFlow fue desarrollado por investigadores e ingenieros del equipo de '*Google Brain*'. Se construye alrededor de un gráfico del cómputo compuesto de un sistema de nodos. Cada nodo representa una operación que puede tener cero o más entradas o salidas. Los valores que fluyen a través de los bordes del gráfico de cómputo se denominan tensores [56].

Los tensores pueden entenderse como una generalización de escalares, vectores, matrices y así sucesivamente. Más concretamente, un escalar puede ser definido como un tensor rango 0, un vector como un tensor rango 1, una matriz como un tensor rango 2 y matrices apiladas en tercera dimensión como tensores de rango 3 [50].

La arquitectura de TensorFlow se maneja por medio de diversas capas, las cuales se pueden observar en la figura 2.8, la cual nos muestra una capa que agrupa las librerías en alto nivel, englobando Python, Tensor Board, C++ y más. La siguiente capa proporciona el núcleo de TensorFlow, el cual contiene una red neuronal de operaciones, el motor de ejecución gráfica y más. Por último se encuentra la capa de las plataformas, basada en los CPUs, GPUs, Android, iOS siendo los principales sistemas operativos.

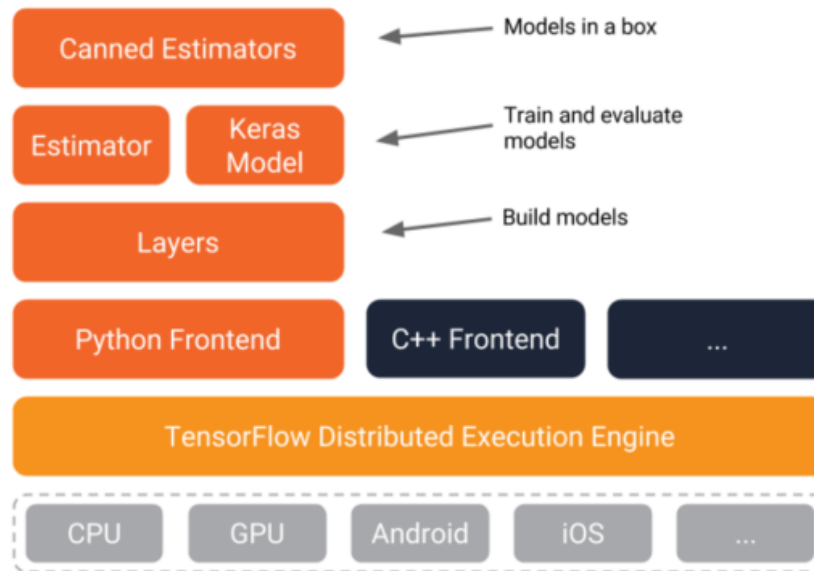


Figura 2.8. Arquitectura de TensorFlow [57].

2.5.3 Keras

Es una API de redes neuronales a alto nivel, escrita en Python y es capaz de ejecutarse en la parte superior de TensorFlow o Theano. El objetivo por el cual fue desarrollada es permitir la experimentación rápida. Ser capaz de pasar de la idea al resultado con el menor retraso posible es clave para hacer una buena investigación [58].

Algunos de los principios por los cuales Keras puede ser elegido como una opción de uso se describen a continuación:

- Cuenta con facilidad de uso ya que al ser un API diseñada para los humanos y no para las máquinas, es centrada en la experiencia del usuario. Keras sigue las mejores prácticas para reducir la carga cognitiva, ofreciendo API's coherentes y sencillas que minimizan el número de acciones del usuario necesarias para casos de uso común y proporcionan comentarios claros sobre el error del usuario.

- Se entiende como un modelo a una secuencia de módulos independientes y completamente configurables los cuales se pueden conectar con la menor restricción posible. Particularmente, las capas neuronales, funciones de costo, optimizadores, esquemas de inicialización, funciones de activación, esquemas de regularización son módulos autónomos, que a su vez se pueden combinar para crear módulos nuevos.

2.6 Traductores de texto en español a texto LSM

Existen diversos trabajos que versan alrededor de la traducción de texto en español a texto en lengua de señas, como los que se describen en la tabla 2.4, en la cual se describen diversos trabajos y se compara la forma en la que cada uno hace su enfoque de traducción.

Tabla 2.4. Estado del arte referente a la traducción de español a lengua de señas.

Artículo	Autor(es)	Año	Descripción
Traductor de español a LSE basado en reglas gramaticales y morfológicas [59].	Sandra Baldassarri, Francisco J Royo Santas	2006	Consiste en un sistema que busca la traducción a lengua de señas española (LSE) por medio de libros, páginas web o cualquier medio que pueda generar un escrito en español, el cual es usado para poder realizar una comparación y poder determinar qué patrones son los que serán empleados en la traducción del LSE para generar las salidas correspondientes. El sistema emplea como base la gramática española y usa transformaciones en las que toma en cuenta las características sintácticas, morfológicas de las palabras y la semántica de los significados, además, el uso de reglas gramaticales y morfológicas es una de las bases que permite la traducción de frases en tiempo real.

<p>Sistema de traducción automática al español cantandioma con un intérprete virtual [60].</p>	<p>Sandra Baldassarri, Eva cerezo y Francisco Royo-santas</p>	<p>2009</p>	<p>En este trabajo se presenta un módulo de traducción automática de frases en español a LSE, considerando características morfológicas de las palabras y la semántica de su significado. El proceso de traducción se desarrolla por medio de cuatro módulos. Primeramente pasa por un analizador morfosintáctico, el cual usa una frase en español como entrada y extrae la información morfológica y sus dependencias sintácticas. Después llega a una transformación gramatical, el cual se basa en la información reunida durante el proceso anterior y al aplicar reglas gramaticales el módulo genera glosas. Luego se ingresan las glosas en un transformador morfológico, el cual revisa los posibles errores que pueda contener en comparación con los términos usados en LSE, así se limpia la gramática obtenida para dar salida a glosas gramaticalmente correctas. Para finalizar el proceso, se traduce en un formato de representación en el módulo de generación de signos con las glosas obtenidas del módulo anterior.</p>
<p>Traducir información de autobuses en lenguaje de señas para personas sordas [61].</p>	<p>V. López Ludeña, C. González Morcillo, J.C. López, R. Barra Chicote, R. Córdoba, R. San Segundo.</p>	<p>2014</p>	<p>El trabajo desarrollado se encuentra basado en dos sistemas principales, de los cuales el primero permite traducir mensajes de texto de paneles de información y el segundo realiza la traducción del español hablado, constituidos ambos sistemas por un traductor de idiomas básicamente de una frase de Word a LSE. La traducción de lenguaje natural es manejada como un módulo el cual tiene una estructura jerárquica que se divide en dos pasos. El primero hace uso de un corpus paralelo el cual maneja un conjunto de oraciones en el idioma origen y sus traducciones al idioma de destino, y el sistema determina si es similar al texto ya definido a traducción por medio de una heurística entre las oraciones, basándose en la distancia con respecto al umbral. Si es menor corresponde a la misma traducción y si es mayor el sistema no genera ninguna salida y se consideran otras estrategias de traducción. La segunda se basa en traducción estadística, el cual es un proceso de traducción basado en frases y un modelo de lenguaje de destino. El modelo de frases ha sido entrenado para realizar un cálculo de alineación de palabras,</p>

			realizar la extracción de frases y sondeo de frases que son calculados por los pares de frases. Con esto se obtuvo una herramienta con nueva versión capaz de traducir del LSE a español.
Creación de recursos para el sistema de traducción automática de textos en español al lenguaje de señas mexicano [39].	Obdulia Pichardo-Lagunas, Bella Martínez Seis.	2015	Este artículo presenta un sistema que está basado en dos módulos: el traductor y el sintetizador de lengua de signos. Las palabras que se usaron pertenecen al diccionario “ <i>Manos con voz</i> ” [62] y “ <i>Mis Manos que hablan</i> ” [63], además de un corpus que contiene cien oraciones en español, las cuales fueron mapeadas teniendo en cuenta la gramática de la estructura del LSE y validadas usando estructuras sintácticas correctas de la gramática del LSM. A grandes rasgos, el sistema toma el texto de origen y lo procesa por medio de la herramienta FreeLing [27], así identifica los elementos en la oración y define si forman parte de su vocabulario o no. Una vez teniendo el vocabulario autenticado, se procesa por medio del módulo sintetizador a la base de datos, haciendo la búsqueda de la estructura gramatical y determinar si está incluida para mostrar como resultado una salida con imágenes que forman la frase en LSM.
Sistema de traducción directa de español a LSM con reglas marcadas [64].	Obdulia Pichardo-Lagunas, Luis Partida-Terrón, Bella Martínez-Seis, Adriana Alvear-Gallegos, Raúl Serrano-Olea.	2016	Se trata de una herramienta de apoyo que, con determinadas reglas marcadas, permite la traducción directa del español a lengua de señas mexicana. El sistema se encuentra formado por diversos módulos, los cuales apoyan en la generación de un diccionario de palabras, realizan la obtención de etiquetas por medio de un análisis de una frase en español y en la construcción de oraciones con reglas básicas definidas en un léxico para cada palabra recabada del LSM y así generar la salida con las señas correspondientes. Para efectos en la obtención del léxico, el sistema hace una comparación entre el diccionario de la RAE y el etiquetado que realiza el analizador FreeLing [27] y así realizar su clasificación. Por último la salida la muestra de acuerdo a la secuencia de imágenes que se obtuvo de acuerdo a la traducción realizada por los módulos descritos anteriormente.

<p>Arquitectura propuesta de reconocimiento de voz para traducción a lengua de señas mexicana mediante un avatar [65].</p>	<p>Otniel Caraveo Carvajal, Ana Luisa Millán Castro, Lic. Beatriz Cota Ponce, María Trinidad Serna Encinas, César Enrique Rose Gómez.</p>	<p>2017</p>	<p>La arquitectura propuesta está basada en un sistema que hace la definición de un mensaje en español a señas mexicanas. Esto lo realiza por medio de una base de datos que se encarga de registrar las señas correspondientes, de modo que, el mensaje que envía la persona hablante llega al diccionario de palabras para realizar la traducción de español a lengua de señas mexicana. El autor destaca la dificultad que conlleva realizar la traducción debido a que no corresponde palabra a palabra los dos lenguajes, y el mayor acervo lingüístico que toma el español provoca la diferencia en las oraciones comparadas. Por lo cual dentro de la arquitectura se incluye un módulo de limpieza que elimina las palabras que no existen en LSM y aplica un algoritmo de semántica para darle coherencia a la estructura de oraciones, logrando una traducción precisa para la persona sorda. Después de realizar el proceso de formulación de oraciones, hace uso de la base de datos que se muestra cómo diccionario de LSM para poder dar la salida correspondiente a la oración.</p>
<p>Arquitectura propuesta de un sistema de apoyo educativo para personas con discapacidad auditiva [66].</p>	<p>Ernesto Darío Barraza Granillo, César Enrique Rose Gómez, Samuel González López, María Trinidad Serna Encinas.</p>	<p>2017</p>	<p>La arquitectura diseñada para el sistema contiene diversos módulos que realizan el trabajo deseado. La estructura que lleva el sistema para realizar la traducción y mostrar una salida se basa en dos módulos: el módulo de procesamiento de texto que recibe un texto con o sin errores y es encargado de darle la estructura al texto leído para entenderlo de una manera correcta. Este módulo da como salida un texto corregido y estructurado que llega al módulo de procesamiento de la información, que se encarga de realizar el procesamiento de la estructura morfológica, sintáctica y semántica de la oración previamente recibida. Con esto se busca dar paso a una plataforma que realice la traducción coherente y correcta de lo que está recibiendo del expositor y dar las salidas correspondientes en tiempo real. El autor enfatiza la importancia de realizar el proceso sintáctico y semántico de la manera correcta, ya que de eso depende el éxito de la traducción y la comprensión por parte del usuario de la plataforma.</p>

<p>Traductor de texto en español a texto LSM usando aprendizaje profundo.</p>	<p>Juan Carlos Hernández Cruz, César Enrique Rose Gómez, Samuel González López, Ana Luisa Millán Castro, María Trinidad Serna Encinas.</p>	<p>2019</p>	<p>El desarrollo del traductor de texto en español a texto LSM fue pensado para emplearse como una herramienta capaz de adaptarse en una plataforma para apoyo educativo a la comunidad de Sordos, esto con la finalidad de poder realizar las traducciones de la manera más efectiva posible. El esquema del traductor fue diseñado combinando modelos tradicionales (como lo es un autómata o traducción usando reglas) con aprendizaje profundo, una técnica que ha generado muy buenos resultados empleándolo en las máquinas de traducción en los últimos años. El objetivo es desarrollar un modelo de traducción que pueda obtener traducciones efectivas y a su vez logre ir moldeándose por medio de la alimentación de un corpus LSM. La diferencia de este modelo radica en la implementación de redes profundas, ya que ningún traductor de lengua de señas ha trabajado con las mismas.</p>
---	--	-------------	--

El estado del arte nos presenta el uso de los métodos tradicionales como lo son las traducciones basadas en reglas y algunos con traducciones estadísticas para la realización de traducciones con sistemas de lengua de señas, sin embargo, se realizara la aplicación del redes neuronales profundas para la elaboración de este trabajo de tesis por los resultados que se han presentado en la implementación de traductores. Algunas de las más empleadas en la actualidad y que brindan mejores resultados son las redes recurrentes, que serán trabajadas con ayudas de Keras y TensorFlow como herramientas para la generación de las mismas.

CAPÍTULO 3: Análisis y diseño

“El verdadero significado de las cosas se descubre al tratar de decir lo mismo con otras palabras”.

– Charles Chaplin.

Dentro del estudio que se obtuvo con el estado del arte presentado en el capítulo 2, se puede observar la dificultad en la que se encuentra la recolección y presentación de información referente a texto LSM, lo cual expone la relevancia de efectuar diversos métodos experimentales para conseguir realizar las traducciones esperadas del trabajo elaborado. Con lo mencionado anteriormente, se plantea una arquitectura con la que se trabajará el texto en español para poder realizar las traducciones que serán mostradas en la plataforma de apoyo educativo [66].

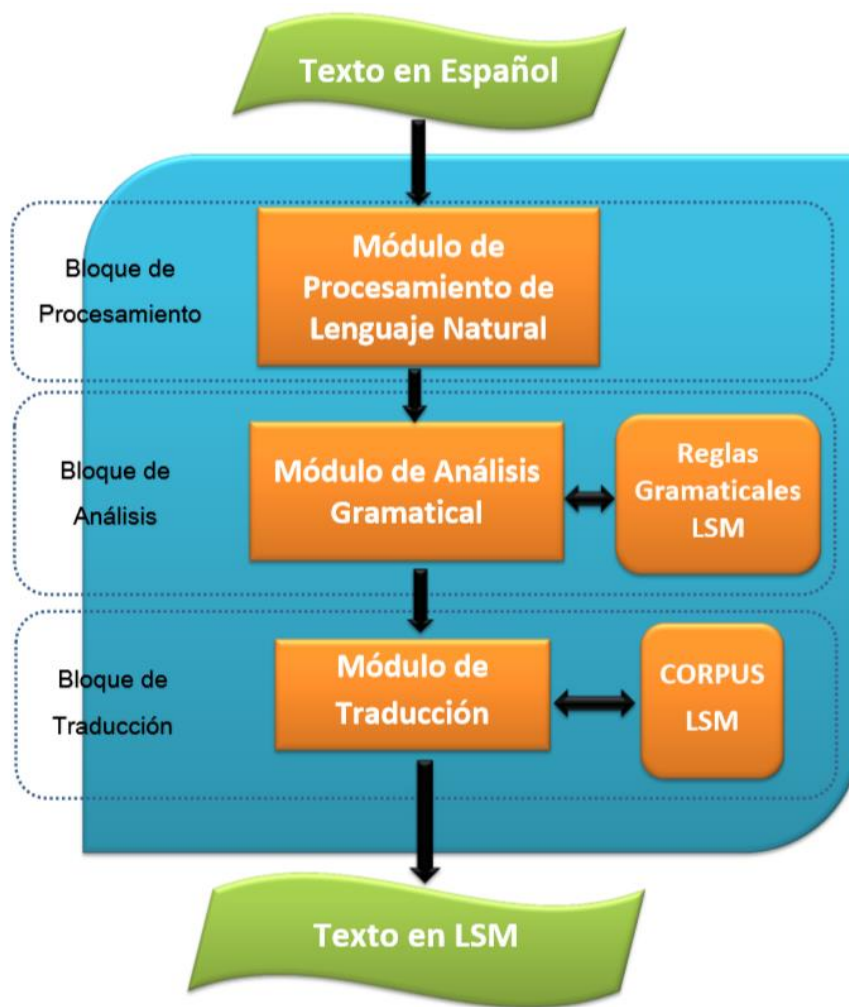


Figura 3.1. Arquitectura del traductor de español a texto LSM.

En la figura 3.1, se muestra la arquitectura del sistema desarrollado, en donde se muestran los diferentes módulos que la conforman, como lo son: el módulo de Procesamiento de Lenguaje Natural (PLN), el módulo de análisis gramatical y por último el módulo de traducción, los cuales se describen detalladamente en los apartados del 3.1 al 3.3.

Dentro del desarrollo y funcionalidad, se busca que la arquitectura reciba un texto en español y este sea el punto de partida para realizar el procesamiento de lenguaje natural, en donde el módulo obtendrá una estructura lematizada bien definida de dicha oración y a continuación la envía al módulo de análisis gramatical, este a su vez, con ayuda de reglas gramaticales definidas del LSM, busca obtener una traducción lo más parecida a la traducción exacta que se desea obtener. Esto da como resultado la instancia que recibe el módulo de traducción de texto, quien por medio de diversas técnicas de traducción (definidas en el punto 3.3) y, creando y haciendo uso de un Corpus de LSM, se busca que realice la traducción precisa de texto en glosa LSM para la plataforma de apoyo educativo.

En seguida se describen particularmente cada uno de los módulos que conforman el sistema de traducción y en algunos de ellos los componentes que la conforman internamente.

3.1 Módulo de procesamiento de lenguaje natural

Esta etapa del sistema se basa en realizar la extracción de cada una de las partes del texto recibido de la plataforma, en un módulo que trabaja los diferentes niveles en los que se encuentra la oración dentro del PLN y mostrando una lematización

con ayuda de la herramienta FreeLing [27]. Los niveles bajo los que se trata el texto son el morfológico, sintáctico, semántico, pragmático y discurso por último.

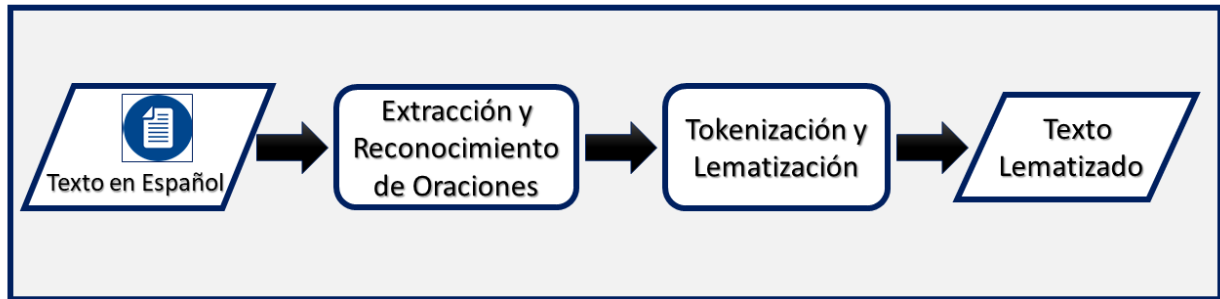


Figura 3.2. Módulo de PLN

La figura 3.2 nos muestra como es el proceso interno del módulo, el cual realiza la recuperación del texto que se va a tratar, y con ayuda de la herramienta FreeLing, lleva a cabo un proceso de etiquetado para cada una de las palabras que se reconocen en el texto, obteniendo las partes de la oración para cada una de las sentencias quedando tokenizada cada una y así obteniendo su lematización correspondiente, de esta manera se pueda generar la instancia que se va a procesar en el siguiente módulo.

El texto que se obtenga de la lematización será visualizado en formato JSON y/o XML donde cada una de las oraciones en español muestra, entre varias características, su lema y etiqueta correspondiente a cada una de las palabras reconocidas por FreeLing y esta es la forma en la que se busca trabajar con la oración.

Algoritmo 3.1. Procesamiento de texto para extracción de lemas.

```
Inicio  
  texto = abrir(ruta_texto_español)  
  parámetros = "parametros"; url = "url_servicio_freeling"  
  petición_servicio = post (url, texto, parametros)  
  objeto = convertir a formato JSON(petición_servicio)  
  
  for oración en objeto: #Extracción y reconocimiento de oraciones  
    for palabra en oración:  
      texto_lematizado = palabra  
    end for  
  end for  
  regresar texto_lematizado  
Fin
```

El algoritmo 3.1 para el procesamiento de lenguaje natural y el primer módulo, que inicia con la apertura de un archivo de texto y lo guarda como texto dentro del programa, después se agregan los parámetros y la ruta del servicio de FreeLing. Con esto se tienen los elementos para realizar la petición al analizador de texto quien regresa el texto lematizado como un objeto.

Por último se hace la extracción y el reconocimiento de las oraciones para poder dar salida al texto lematizado de la palabra reconocida.

3.2 Módulo de análisis gramatical para LSM

Dentro de este módulo, lo que se busca conseguir es el acercamiento a la forma gramatical más próxima sobre la traducción real, convirtiendo el texto lematizado que se genera en el módulo anterior y, con apoyo de las reglas gramaticales de la LSM que se obtuvieron del trabajo de tesis doctoral de la Dra. Miroslava Cruz [7], formular una oración en glosa LSM.

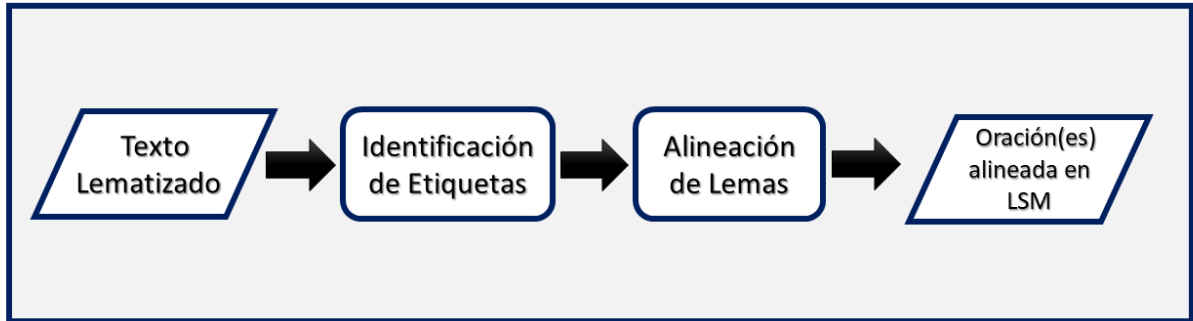


Figura 3.3. Módulo de análisis gramatical.

El procesamiento que se lleva a cabo dentro del módulo de análisis gramatical se muestra en la figura 3.3, es inicializado por el texto lematizado, con esta instancia se busca identificar cada una de las etiquetas de las palabras que contiene la oración reconocidas en el módulo anterior, y con base en la gramática de la LSM, se busca obtener una alineación estadística de los lemas etiquetados, de tal forma que se pueda lograr generar una oración con las características gramaticales de la lengua a obtener (en este caso de la glosa LSM).

La gramática usada para la identificación y alineamiento de lemas, haciendo referencia desde el uso de la fonología hasta la sintaxis usada en las oraciones de la LSM, es mostrada en el trabajo [7] de la siguiente manera:

1. Sujeto - Verbo - Objeto
2. Tiempo - Sujeto - Verbo - Objeto
3. Tiempo - Lugar - Sujeto - Verbo - Objeto - Adverbio
4. Tiempo - Lugar - Sujeto - Verbo - Objeto - Negación
5. Tiempo - Lugar - Sujeto - Verbo - Objeto - Pregunta

En la cuestión de entidades nombradas que sean encontradas dentro de la oración, se va a proceder a marcarlas como palabras con “deletreo de dedos” y la palabra será separada por espacios para reconocer dicho elemento.

Algoritmo 3.2. Procesamiento de análisis gramatical.

```
Inicio
  Crear variables sentencia[], oracion_limpia[].
  for pos en texto_lematizado:
    if pos es igual a 'verbo':
      entonces sentencia = 'form + lemma + ctag + pos + tense + person'
    else if pos es igual a 'sustantivo' y tipo es igual a 'propio':
      entonces sentencia = 'form + lemma + ctag + pos + gen'
    else:
      entonces sentencia = 'form + lemma + ctag + pos'
    end if
  end for

  for pos en sentencia:
    elemento = división de pos
    if elemento es diferente de 'articulo':
      entonces oracion_limpia = elemento
    end if
  end for
  regresar texto_lematizado
Fin
```

En el algoritmo 3.2 se busca obtener la alineación de los lemas, usando las reglas gramaticales y realizando una limpieza del texto lematizado, dejando cada uno de los atributos que se van a usar para el módulo de traducción. Al obtener este filtrado se hace un barrido para descartar las palabras que son artículos, ya que no son parte de la gramática de la LSM.

3.3 Módulo de Traducción

Actualmente, en el estado del arte referente a las técnicas de traducción empleadas en las lenguas de señas que se han utilizado y han arrojado resultados satisfactorios hace referencia al uso de los sistemas basados en reglas y los traductores estadísticos, y no solo el empleo de cada uno de estos, sino una combinación en ellos que permite que las traducciones alcancen resultados completamente aprobatorios y hasta el momento los más altos por arriba del 98% de eficiencia.

Para los traductores, hablando particularmente de traducciones a la LSM, los modelos que se utilizan son los sistemas basados en reglas, debido a la cantidad de información con la que se cuenta acerca de la lengua de señas, aun no se ha podido trabajar con datos estadísticos.

En este módulo se emplea una combinación de dichas técnicas de traducción, por una parte se ejecuta un autómata, que con ayuda de las reglas generadas busca la traducción (modelo basado en reglas), y por otra parte la red profunda, que por medio del aprendizaje previo realizará la búsqueda de la mejor combinación para mostrar (modelo basado en aprendizaje profundo).

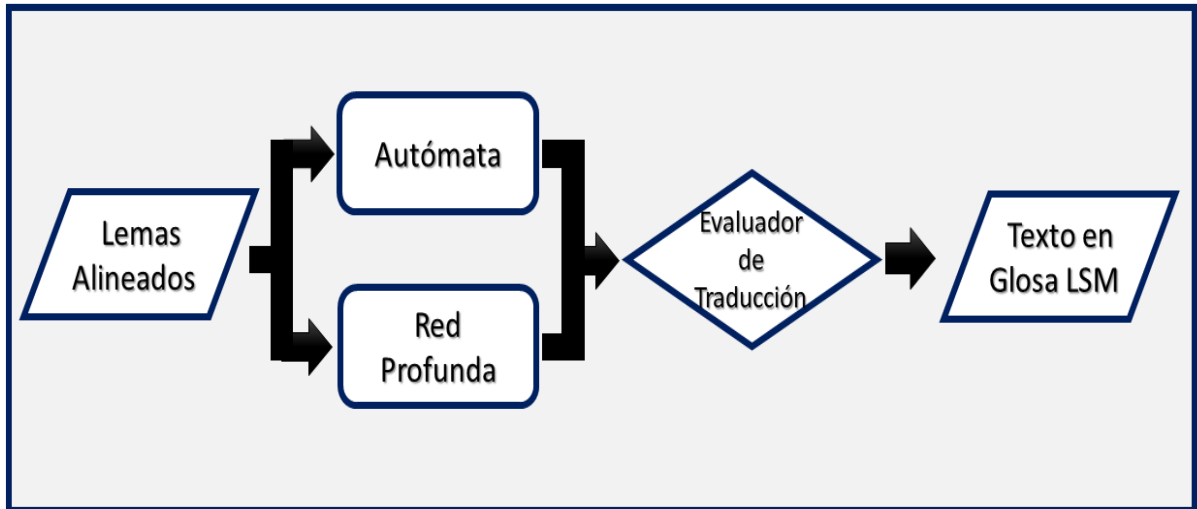


Figura 3.4. Módulo de traducción.

La figura 3.4 nos muestra el proceso detallado en el que es trabajada la instancia que proporciona el módulo de análisis, donde a partir del conjunto de lemas previamente alineados, se busca realizar la traducción en dos diferentes modelos. El primer modelo busca realizar una traducción por medio de un autómata, que consiste en un modelo basado en reglas obtenidas del módulo anterior, este módulo va a permitir realizar la traducción lo más apegado a la gramática que se está trabajando y de esta forma obtener una traducción correcta.

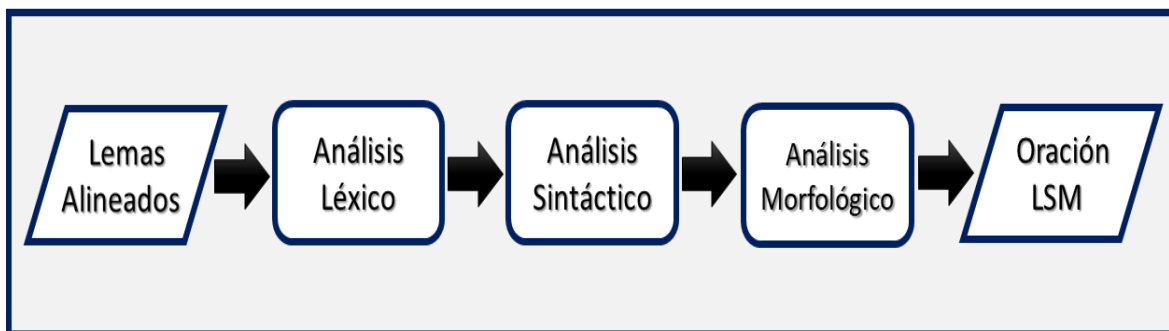


Figura 3.5. Modelo de traducción con un autómata.

Se puede observar el proceso que lleva el autómata detalladamente en la figura 3.5, donde recibe los lemas alineados y realiza un proceso de análisis léxico para verificar la lematización y tokenización previamente realizada, después de eso se procede a un análisis sintáctico

Además se busca realizar la traducción con métodos de aprendizaje profundo, que con ayuda de una red recurrente y un corpus de LSM se consigue alimentar y entrenar, y de esta manera encontrar el modelo que mejor se adapte a las reglas gramaticales y así realizar las traducciones. Este modelo se basa en una LSTM y el uso de Seq2Seq de Keras para poner en práctica esta forma de traducción y tener una comparación de cual modelo es el que mejor realiza la traducción esperada.

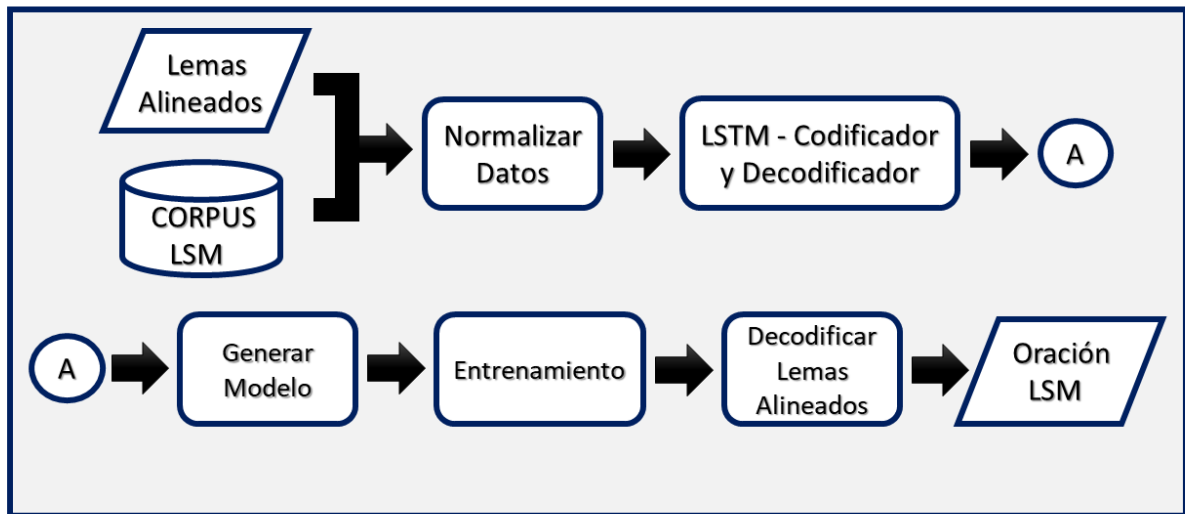


Figura 3.6. Modelo de traducción con red profunda.

Al encontrar cada uno de los modelos su traducción, se busca por medio de una heurística hacer una comparación en el módulo de evaluación y determinar cuál es

el texto en glosa LSM que mejor resultado obtiene y al final ser enviado para que se presente como el texto en glosa LSM para la plataforma de apoyo educativo.

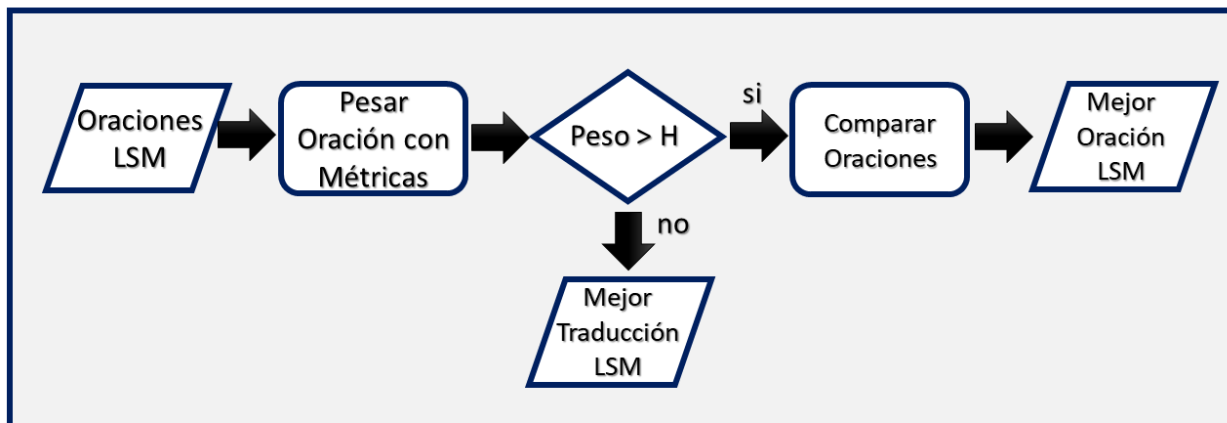


Figura 3.7. Modelo de evaluación.

3.4 CORPUS español - lengua de señas

Los recursos que existen para el trabajo de traducción de idiomas de lenguas escritas a idiomas de lengua de señas son bastante escasos, y específicamente sobre el español a LSM es un trabajo de investigación que se está iniciando, por lo tanto se tuvieron que tomar diversas medidas para poder realizar el desarrollo y la experimentación del proyecto.

Para efectos de experimentación se obtuvo un corpus de español a texto en glosa LSE [69] que se desarrolló utilizando seis textos diferentes con diversos temas como alojamiento estudiantil, historia y guerra, un día en la vida de un perro, la población y los problemas demográficos, la violencia doméstica, y un día en la vida de una niña. Los textos uno y dos son textos sencillos, los textos tres y cuatros tienen una

dificultad media, y los textos cinco y seis cuentan con la dificultad más alta de entendimiento.

Además de esto, la estructura con la que se encuentra desarrollado el corpus es mostrado en tres diferentes versiones descritas de la siguiente manera: (A) es una oración en versión española obtenida de la fuente, (B) es una traducción al LSE obtenida de la oración A, (C) es una traducción al español obtenida de la oración B.

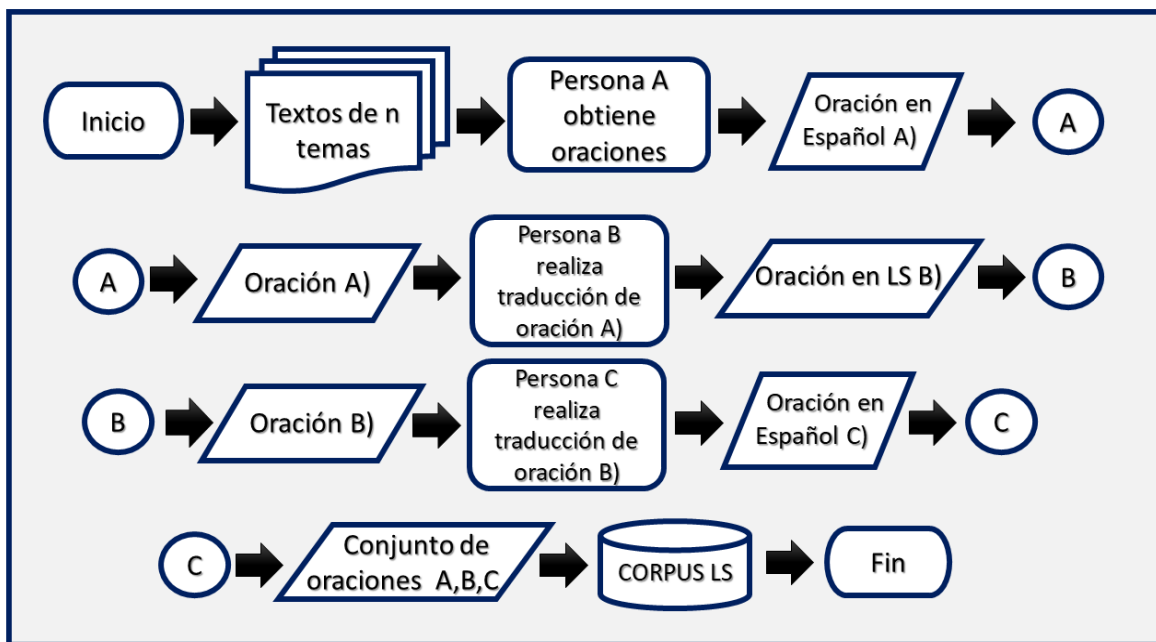


Figura 3.8. Proceso de desarrollo del corpus LSE [69].

Algunos de los ejemplos de oraciones que se pueden observar dentro del corpus empleando la técnica descrita en la figura 3.8 son presentados en la tabla 3.1:

Tabla 3.1. Ejemplos de oraciones del corpus.

Oración A (Fuente)	Oración B (Traducción español a LSE)	Oración C (Traducción LSE a español)
Los que tienen más suerte son aquellos que tienen hermanos o amigos que ya viven en Sevilla y que les ayudan a buscar piso.	ALGUNOS SUERTE HAY AMIGO O HERMANO, ETCÉTERA, ESOS PERSONA++ APOYARME PARA BUSCAR PISO.	Algunos tienen suerte y tienen amigos o hermanos que les apoyan para buscar piso.
Esta es la solución que prefieren los estudiantes.	ESE, ALQUILER PISO GUSTAR MÁS QUIÉN, PERSONA+ ESTUDIANTE+.	El alquiler de piso les gusta más a los estudiantes.
Los padres se sienten más tranquilos y relajados con la residencia	MOTIVO PADRE-MADRE TRANQUILO, RELAJADO TEMA RESIDENCIA	Porque se sienten tranquilos y relajados con la residencia
La guerra es tan vieja como la humanidad.	GUERRA EDAD IGUAL HUMANIDAD IGUAL+.	La guerra tiene la misma edad que la humanidad.
Cada mujer de los países de la Unión Europea tiene 1,4 hijos.	UN MUJER PAÍS+ UNIÓN EUROPEA EN-TOTAL HIJO 1,4 HIJO UN MUJER.	En los países de la Unión Europea, en total, cada mujer tiene 1,4 hijos.
Ana ya sabe leer y escribir. Es muy lista.	ÉL MUJER ("Ana") SABER LEER, ESCRIBIR, MUY-INTELIGENTE.	Ana sabe leer y escribir, es muy inteligente.

CAPÍTULO 4: Integración de Herramientas Computacionales

*“El lenguaje sirve no solo para expresar el pensamiento, sino para hacer posibles pensamientos que no podrían existir sin él”.
– Bertrand Russell.*

En el capítulo anterior se observa la estructura y los requerimientos que lleva el traductor para su desarrollo. En este capítulo se muestra la integración de los

módulos de procesamiento de lenguaje natural, la alineación gramatical y el módulo de traducción respectivamente de cada uno de los modelos de traducción a usar. Se muestran los diferentes métodos de traducción usados desde el lenguaje Python, con ayuda de las librerías Freeling, Keras y Tensor Flow, con las cuales fueron desarrollados cada uno de los módulos.

4.1 Entorno de desarrollo

En la actualidad, uno de los problemas más comunes al hacer uso de modelos de aprendizaje computacional es contar con el equipo de cómputo adecuado que será utilizado para realizar las implementaciones de las redes neuronales.

Para efectos de este proyecto se trabajó con un equipo que fue proporcionado por el laboratorio de sistemas inteligentes en el Instituto Tecnológico de Hermosillo, donde fueron desarrollados tanto el entrenamiento como las pruebas. En la tabla 4.1 se describen las características con las que cuenta el equipo.

Tabla 4.1. Características del equipo de cómputo del proyecto.

Sistema Operativo	Ubuntu 18.0 Lite x64 bits
--------------------------	---------------------------

Procesador	Intel(R) Core(TM) i7-4785T CPU 2.20 GHz
Memoria RAM	16.0 GB
Disco Duro	980 GB SSD
Tarjeta Gráfica	Intel® HD Graphics 520 (Skylake GT2)

Una vez teniendo el equipo de cómputo a usar, se realizan las instalaciones de las librerías necesarias para el desarrollo del proyecto. Para el uso de Python en un entorno gráfico amigable se instaló Anaconda Navigator quien permite el uso de entornos gráficos virtuales con las características requeridas para el desarrollador. Al instalar Anaconda Navigator se preparó el entorno virtual desde Jupyter Notebook, que nos permite manejar las librerías necesarias como lo es FreeLing, Keras, Pandas, entre otras.

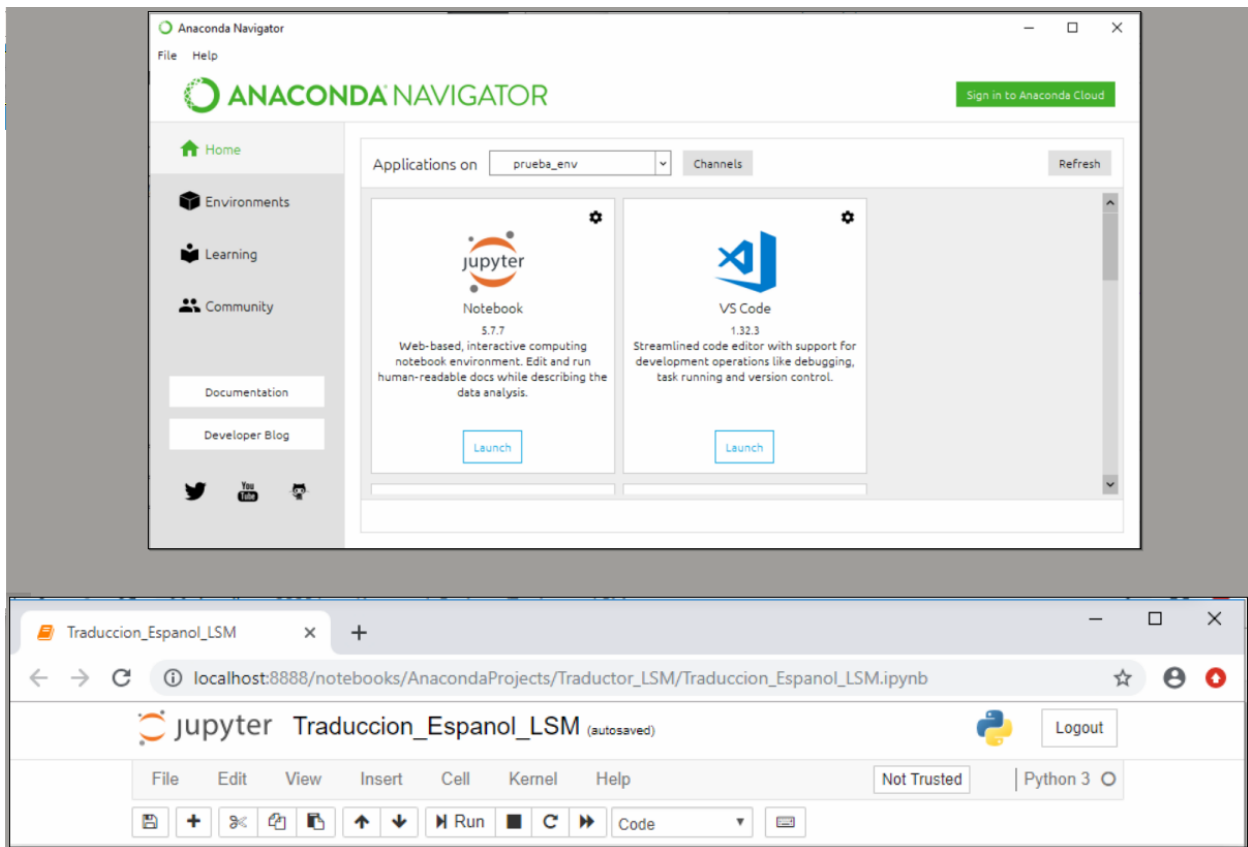


Figura 4.1. Anaconda Navigator y el entorno de desarrollo.

Una vez generado el entorno de desarrollo en Jupyter Notebook, se puede ingresar al mismo desde la ventana de Anaconda, en donde nos abre el entorno de desarrollo creado desde el navegador predeterminado, direccionando al servidor web local de nuestro equipo (<http://localhost:8888>) lo cual se muestra en la figura 4.1.

4.2 Algoritmo para el procesamiento de lenguaje natural

La primera entrada será recibida y leída desde un archivo de texto en el módulo diseñado donde se usa un algoritmo para la extracción de lemas del texto en español que se recibe para el trabajo de traducción.

Este algoritmo funciona con base a la utilidad FreeLing, que desde el lenguaje de programación Python, hace uso de la librería 'pyfreeling' el cual emplea un analizador para poder generar el lematizado de la o las oraciones reconocidas y de esta manera genera una cadena de texto en formato XML que contiene el token de la oración, el lema de la palabra, la etiqueta y una variación corta de la misma etiqueta de la palabra, además de la parte de la oración a la que corresponde dicha palabra con diferentes variaciones adicionales entre sustantivos, adjetivos y verbos respectivamente a cada uno.



Figura 4.2. Fragmento de código de librerías para procesamiento de lenguaje.

Como se muestra en la figura 4.2, se importan las librerías para el uso de FreeLing en Python y desde el archivo 'config' se ajustan todos los parámetros que corresponden al lenguaje a trabajar, y el cual contiene las rutas específicas del analizador.

```

#Lectura del archivo
archivo = open('/home/jc/myfile.txt','r')
texto = archivo.read()
archivo.close()

# Analizar el texto y obtener los lemas
analizador = Analyzer(config=config_file, lang='es')
texto_lematizado = analizador.run(texto.encode())

```

Figura 4.3. Fragmento del código de lectura del archivo y el analizador.

La figura 4.3 nos muestra la lectura del archivo para extraer el texto que se trabajará y se pasa al método analizador, que con los parámetros del archivo de configuración, realizará la tokenización del texto reconocido del archivo y lo guardará en un parámetro llamado *texto_lematizado* para su uso posterior del siguiente módulo.

```

print(etree.tostring(texto_lematizado))

```

```

b'<sentences>b'\<sentence id="1">\n <token id="t1.1" begin="0" end="4" form
="Esto" lemma="este" tag="PD00S00" ctag="PD" pos="pronoun" type="demonstrative"
num="singular">\n </token>\n <token id="t1.2" begin="5" end="7" form="es" l
emma="ser" tag="VSIP350" ctag="VSI" pos="verb" type="semiauxiliary" mood="indic
ative" tense="present" person="3" num="singular">\n </token>\n <token id="t
1.3" begin="8" end="11" form="una" lemma="uno" tag="DI0FS0" ctag="DI" pos="dete
rminer" type="indefinite" gen="feminine" num="singular">\n </token>\n <toke
n id="t1.4" begin="12" end="18" form="prueba" lemma="prueba" tag="NCF5000" ctag
="NC" pos="noun" type="common" gen="feminine" num="singular">\n </token>\n
<token id="t1.5" begin="19" end="21" form="en" lemma="en" tag="SP" ctag="SP" po
s="adposition" type="preposition">\n </token>\n <token id="t1.6" begin="22"
end="29" form="Francia" lemma="francia" tag="NP00G00" ctag="NP" pos="noun" type
="proper" necclass="location" nec="LOC">\n </token>\n <token id="t1.7" begin
="29" end="30" form="." lemma="." tag="Fp" ctag="Fp" pos="punctuation" type="pe
riod">\n </token>\n</sentence>\n'\</sentences>'

```

Figura 4.4. Visualización del texto lematizado.

La forma en la que se visualiza la lematización del texto puede observarse dentro de la figura 4.4, que a grandes rasgos nos muestra las características de la estructura de la oración, mencionando dentro de las más importantes el token de la palabra, la raíz, el lema, la etiqueta de la oración, que tipo de palabra es dentro de la oración, género y número entre otras.

4.3 Algoritmo para la alineación de lemas

Dentro del procesamiento de texto se incluyen tareas tanto de limpieza como de selección de ciertas características, que son mencionadas anteriormente, según corresponda a cada una de las partes de la oración reconocidas.

Cuando la oración llega al módulo de análisis gramatical, se procede a detectar cada uno de las partes de la oración para identificar y realizar la alineación según corresponda la gramática SVO de la lengua de señas mexicana.

Inicialmente se desglosa cada una de las palabras y se realiza el reconocimiento de palabras que fungen como sujeto (los sustantivos), los objetos (adjetivos, adverbios) y los verbos. Dentro de la gramática se descartan las palabras identificadas como artículos, ya que no son usados dentro de la gramática correspondiente a la glosa del LSM.

Módulo de Análisis Gramatical

```
sentencia = []

for ch in texto_lematizado[0]:
    #print(ch.attrib)
    if ch.attrib['pos'] == 'verb':
        sentencia.append(ch.attrib['form'] + '###' + ch.attrib['lemma'] + '###')
    else:
        sentencia.append(ch.attrib['form'] + '###' + ch.attrib['lemma'] + '###')

oracion_limpia = []

for x in sentencia:
    r=x.split('###')
    if r[3] != 'determiner':
        #print(r)
        oracion_limpia.append(r)
```

Figura 4.5. Fragmento de código de la limpieza gramatical.

En la figura 4.5 se presentan dos fragmentos del código que determina cuáles serán las características que se van a seleccionar después del desglose que se realiza en el procesamiento del módulo anterior, según corresponda en la parte del etiquetado de la oración.

Además de eso se reduce a una oración limpia que no contendrá los artículos, mencionado anteriormente, ya que no forman parte a nivel sintaxis de la glosa LSM.

4.4 Algoritmo para la traducción

En esta sección se muestran específicamente cada una de las implementaciones que se utilizaron para el módulo de traducción, así como cada uno de los resultados que se obtienen para poder realizar una comparación y evaluar visualmente las salidas que se están obteniendo.

4.4.1 Modelo usando autómatas

El autómata presentado en la figura 3.5 fue diseñado y construido en base a la estructura gramatical de la glosa LSM estudiada en la tesis presentada por la Dra. Miroslava Cruz [7], quien menciona el uso de Sujeto Verbo Objeto en la sintaxis que se maneja en la estructura.

Partiendo de esto, y trabajando lo obtenido desde el módulo de procesamiento de lenguaje natural y el módulo de análisis gramatical que nos dejan con una estructura semejante a la que se busca en esta traducción, se busca hacer los ajustes necesarios para poder obtener la oración en glosa LSM.

Autómata

```
1 glosa_lsm_a = ''
2
3 for x in oracion_limpia:
4     if x[3] == 'verb':
5         tiempo = tense(x[4])
6         glosa_lsm_a = glosa_lsm_a + x[1] + ' ' + tiempo + ' '
7     else:
8         glosa_lsm_a = glosa_lsm_a + x[1] + ' '

1 print('Texto Original: ' + texto)
2 print('Glosa LSM: ' + glosa_lsm_a)

Texto Original: El perro verde regalo flores a Maria ayer.
Glosa LSM: perro verde regalo flor a maria ayer .
```

Figura 4.6. Fragmento de código de la traducción con un autómata.

En la figura 4.6 podemos observar una parte del código del autómata que nos hace el ajuste del texto que se recibe por parte del módulo anterior. La estructura que se maneja es simple y tanto el léxico y la sintaxis son revisados en los módulos previos al ingreso del autómata, donde la revisión más exhaustiva por parte del autómata se realiza en cuestión a la semántica.

Al final de la figura 4.6 se puede observar el texto inicial y el texto en glosa LSM como resultado de todo el procesamiento previo.

4.4.2 Modelo usando redes profundas

El desarrollo de la red fue dividido en dos partes, la primera consta del entrenamiento con el conjunto de datos que se tiene del 'CORPUS LSM'.

Entrenamiento

Al llegar la información del análisis gramatical que son los lemas alineados al módulo de la red profunda, se comienza por realizar la preparación y limpieza de los datos para poder evitar que se tengan elementos no deseados que hagan ruido por parte de la oración que se recibe.

Limpieza de Datos

```
lines.esp=lines.esp.apply(lambda x: x.lower())
lines.LSE=lines.LSE.apply(lambda x: x.lower())

# Take the Length as 50
lines.esp=lines.esp.apply(lambda x: re.sub(" ", "", x)).apply(lambda x: re.sub(",", " COMMA", x))
lines.LSE=lines.LSE.apply(lambda x: re.sub(" ", "", x)).apply(lambda x: re.sub(",", " COMMA", x))

exclude = set(string.punctuation)
lines.esp=lines.esp.apply(lambda x: ''.join(ch for ch in x if ch not in exclude))
lines.LSE=lines.LSE.apply(lambda x: ''.join(ch for ch in x if ch not in exclude))

remove_digits = str.maketrans('', '', digits)
lines.esp=lines.esp.apply(lambda x: x.translate(remove_digits))
lines.LSE=lines.LSE.apply(lambda x: x.translate(remove_digits))
```

	esp	LSE
0	con el mes de septiembre para muchos estudiant...	toca mes septiembre persona estudiante empezar...
1	casi la mitad de los estudiantes de la univers...	casi mitad persona estudiante universidad sevi...
2	el primer y principal problema con que se encu...	primero problema cual COMMA persona estudiante...
3	¿donde vivir es la pregunta que a muchos estud...	vivir donde muchasveces porlanoche cls "perso...

Figura 4.7. Fragmento de código de la limpieza de datos.

Como se muestra en la figura 4.7, con ayuda de los métodos *lower*, *sub* y *maketrans* de Python lo primero a realizar es colocar en minúsculas todos los caracteres de la oración, se eliminan las comillas, las comas (',') se sustituye por la palabra reservada 'COMMA', los caracteres especiales y los números de igual manera son eliminados del texto.

Para comenzar a trabajar los datos, se agregan dos tokens para definir el inicio y el final de la oración destino, lo que ayudará a identificar el procesamiento de los datos dentro de la red profunda. Después se continúa con la identificación del vocabulario tanto para el texto en español como para la glosa en LSM, calculando el tamaño del vocabulario y la longitud de la oración más grande que se tiene tanto para el español como para el LSM.

```
lines.LSE = lines.LSE.apply(lambda x : 'START_ '+ x + ' _END')
```

```
all_esp_words=set()
for esp in lines.esp:
    for word in esp.split():
        if word not in all_esp_words:
            all_esp_words.add(word)

all_lse_words=set()
for lse in lines.LSE:
    for word in lse.split():
        if word not in all_lse_words:
            all_lse_words.add(word)
```

```
input_token_index = dict(
    [(word, i) for i, word in enumerate(input_words)])
target_token_index = dict(
    [(word, i) for i, word in enumerate(target_words)])
```

Figura 4.8. Fragmento de código de generación de datos.

Finalmente se crea un diccionario de tokens tanto del origen como del destino, como se muestra en la figura 4.8, que ayudará a la identificación de palabras en el entrenamiento.

El siguiente paso es generar la red profunda que consta de una red recurrente construida con un modelo codificador-decodificador implementado con la librería Keras.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, None)	0	
input_2 (InputLayer)	(None, None)	0	
embedding_1 (Embedding)	(None, None, 50)	48250	input_1[0][0]
embedding_2 (Embedding)	(None, None, 50)	33700	input_2[0][0]
lstm_1 (LSTM)	[(None, 50), (None, 20200		embedding_1[0][0]
lstm_2 (LSTM)	[(None, None, 50), (20200		embedding_2[0][0] lstm_1[0][1] lstm_1[0][2]
dense_1 (Dense)	(None, None, 674)	34374	lstm_2[0][0]

=====
 Total params: 156,724
 Trainable params: 156,724
 Non-trainable params: 0

Figura 4.9. Modelo codificador-decodificador construido.

El modelo fue definido por medio de una red de memoria a largo plazo (LSTM) para las entradas del codificador y otra red LSTM para las entradas el decodificador. Además de esto se generaron los *embeddings* tanto para codificador como el decodificador que ayudaran con el manejo del vocabulario dentro del procesamiento en la LSTM.

De igual forma se muestra la implementación de la red densa que nos ayudará en el trabajo de la realización de predicciones en las salidas que se tendrá del modelo codificador-decodificador. El modelo generado puede ser observado con cada una de sus características en la figura 4.9.

Finalmente, generamos la secuencia de salida invocando un bucle de oraciones decodificadas la cual genera el proceso que se realizó en el entrenamiento con la oración de entrada que manda la validación.

```
for seq_index in [33, 123, 126, 128]:
    input_seq = encoder_input_data[seq_index: seq_index + 1]
    decoded_sentence = decode_sequence(input_seq)
    print('-')
    print('Input sentence:', lines.esp[seq_index: seq_index + 1])
    print('Decoded sentence:', decoded_sentence)
```

Figura 4.10. Fragmento de código de validación de oraciones.

En la figura 4.10 podemos observar un fragmento del código para visualizar el resultado de las predicciones con el modelo construido y entrenado las oraciones del conjunto de validación.

CAPÍTULO 5: Pruebas y análisis de resultados

“Si hablas con un hombre en un idioma que él entiende, eso va a la cabeza. Si le hablas en su propio idioma, eso va a su corazón”. – Nelson Mandela.

En este capítulo se presentan las pruebas y el análisis de los resultados que se obtuvieron en la integración del sistema, donde se llevaron a cabo diferentes tipos de pruebas. Primeramente se muestran los tipos de evaluaciones de las traducciones considerando las métricas que se emplearon, seguido de la preparación de los datos para la experimentación y la ejecución de la misma, y posteriormente los resultados que se generaron en cada uno de los experimentos que se realizaron.

5.1 Formas de evaluación de las traducciones

A continuación se mostrarán las métricas empleadas para realizar la evaluación del desempeño en las traducciones que se están generando por medio del traductor como evaluación automática. También se ha realizado una evaluación manual con el propósito de que el traductor sea comparado con expertos en el área de traducción de glosa LSM y poder mejorar las traducciones, además de enriquecer el corpus con más oraciones.

5.1.1 Métricas

En la evaluación de la calidad de las traducciones automáticas generadas, se usaron las métricas BLEU y WER las cuales serán explicadas.

BiLingual Evaluation Understudy (BLEU) [72]

Es una de las métricas más utilizadas para la evaluación automática de las máquinas de traducción propuesta por [67, 72]. Esta métrica se basa en el cálculo de la media de precisión para cada n-grama entre la traducción origen y la traducción destino, asignando un peso a cada n-grama y poder comparar con la traducción humana. El método tiene un problema con las oraciones cortas, por lo cual se incorpora un factor de penalización denominado BP, que evita dar demasiado peso a los n-gramas cortos. La medición de la traducción evaluada oscila entre 0 y 100, en donde 100 corresponde a una traducción correcta.

Word Error Rate/Tasa de error de palabra (WER) [67]

Comúnmente empleada en sistemas de reconocimiento de voz, esta métrica de igual manera permite realizar la evaluación de un sistema de traducción automática. La métrica emplea una distancia Levenshtein [67] que es definida como el número de pasos mínimo de edición entre las palabras de una oración, tomando en cuenta las inserciones, eliminaciones y sustituciones que son aplicadas después de obtener una traducción destino para poder ser medida como una traducción correcta. Una desventaja es que se basa en una única traducción válida, por lo cual permite únicamente observar un acercamiento a una forma de evaluación de la traducción. La medición de evaluación de esta métrica es en base a el porcentaje de errores entre el número total de palabras de la traducción, lo cual arroja un porcentaje entre 0% y 100%.

$$WER = \frac{S + D + I}{N}$$

En la fórmula se presenta la forma de calcular el porcentaje de errores, donde S es para el número de sustituciones, si una palabra de la referencia fue sustituida por una palabra alineada de la hipótesis, D es para el número de eliminaciones, si se elimina una palabra de la referencia, I es para las inserciones realizadas, si se agregó una palabra alineada de la hipótesis y N es para el número de palabras en la hipótesis. Con se mencionó anteriormente se obtiene un error, el cual entre más alto sea, peor será la traducción medida.

5.1.2 Evaluación manual

Una manera de medir la calidad una traducción automática de una forma más exacta es por medio de una evaluación manual. Al no tener una traducción óptima, sino diferentes traducciones válidas, es difícil para un algoritmo realizar dicha evaluación, por lo cual una validación exacta es por medio de un experto humano en el área a evaluar debido a que por su experiencia es capaz de diferenciar entre una traducción correcta y un error en la traducción.

Una desventaja del uso de estas evaluaciones es que sin duda se necesita conseguir a las personas capaces de realizar la evaluación y el tiempo que toma el evaluar cada una de ellas.

Para efectos de evaluación del traductor se realizó una lista de oraciones las cuales se le dieron a varios expertos en el área de la glosa LSM con el fin de tener una

base en la que se pueda evaluar la eficiencia con la que se está presentando la traducción del sistema. Además de esta forma se puede asegurar que los datos que se están ingresando al corpus son válidos.

5.1.3 Evaluación automática

Como solución al tiempo que toma realizar las evaluaciones manuales, se optó por emplear de igual forma evaluaciones automáticas. Con esto se busca encontrar una traducción correcta que sea lo más parecido a obtener una traducción humana, esto realizando comparaciones sobre las traducciones destino que se obtienen del traductor y midiendo las oraciones generadas con ayuda de las métricas descritas en la sección 5.1.1.

Así como en la evaluación manual, también se busca que por medio de este tipo de evaluación en las traducciones se puedan utilizar los resultados obtenidos y así agrandar el corpus de glosa LSM para emplearlo en los diversos métodos de traducción existentes y obtener resultados de forma más eficiente.

5.2 Preparación de los datos de prueba

Para realizar los experimentos se ha tomado en cuenta 3 corpus diferentes que pertenecen al dominio de las traducciones.

El primero es un corpus de idioma español-inglés (spa-eng) tomado de [68] que es un repositorio con corpus de aproximadamente 125 idiomas diferentes, todos con oraciones de idioma origen a idioma destino (el idioma destino de todos los corpus es el inglés) separados por tabulaciones. Cada uno de los idiomas tiene diferente variación de número de oraciones, encontrando los idiomas más conocidos como el

alemán, francés, hebreo, ruso y español dentro de los corpus más grandes (más de 100,000 oraciones por corpus).

La tabla 5.1 nos muestra la información general que contiene el corpus español-inglés, en donde podemos encontrar el número total de oraciones en cada uno de los idiomas, el número de palabras diferentes reconocidas en el total de oraciones existentes y el número de palabras que contiene la oración más extensa.

Tabla 5.1 Información corpus español-inglés.

Español	Oraciones	120,245
	Palabras	44,676
	Mayor longitud en oración	53
Inglés	Oraciones	120,245
	Palabras	22,052
	Mayor longitud en oración	49

Las traducciones que se buscan son del área de la lengua de señas por lo que se encontró uno de los corpus más grandes existentes y que nos sirvió para realizar experimentaciones mostrado en la tabla 5.2, este es un corpus de español a lengua de señas española (LSE) con 229 oraciones escritas en glosa que fueron resultado del trabajo de [69].

La tabla 5.2 cumple con las mismas características que la tabla anterior, muestra el número de oraciones que se contienen en cada lenguaje, el número de palabras diferentes reconocidas y el número de palabras que contiene la oración más grande reconocida.

Tabla 5.2 Información corpus español-LSE.

Español	Oraciones	229
	Palabras	978
	Mayor longitud en oración	52
LSE	Oraciones	229
	Palabras	686
	Mayor longitud en oración	46

Por último, empleando las técnicas de recopilación y elaboración del corpus de lengua de señas española, se buscó realizar un corpus así mismo de lengua de señas mexicano (LSM) el cual sirviera para efectos de futuras investigaciones y poder seguir generando más oraciones e ir trabajando en el mismo. El corpus está conformado por 10 oraciones escritas en glosa con las características que se muestran en la tabla 5.3. La tabla contiene el número de oraciones totales, las palabras diferentes reconocidas en las oraciones totales y el número de palabras que tiene la oración con mayor longitud.

Tabla 5.3 Información corpus español-LSM.

Español	Oraciones	10
	Palabras	13
	Mayor longitud en oración	51
LSM	Oraciones	10
	Palabras	18
	Mayor longitud en oración	45

5.2.1 Obtención de datos de entrenamiento y prueba

Para obtener los diferentes datos de entrenamiento y prueba se usó la función de separación de la librería sklearn, la cual permite realizar la división de los datos según la asignación dada. Cuando se cuenta con muchos datos, se puede efectuar una asignación 70% en entrenamiento y 30% en pruebas para poder tener buen margen en ambos conjuntos de datos.

```
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
```

```
X, y = lines.esp, lines.LSE
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.15)
X_train.shape, X_test.shape

((194,), (35,))
```

```
X_train.to_pickle('Weights_LSE/X_train.pkl')
X_test.to_pickle('Weights_LSE/X_test.pkl')
```

Figura 5.1 Fragmento de código para la separación de los conjuntos de datos.

En la figura 5.1 se muestra la forma en la que se realizó la separación de los conjuntos de datos en el corpus de lengua de señas española (LSE), tomando en cuenta que los datos no son bastos, se asignó una separación de 85% para los datos de entrenamiento y 15% para los datos de prueba. Adicional a la separación de los datos, al ser una prueba aleatoria se guarda cada uno de los conjuntos para futuras comparaciones en la experimentación.

Para cada uno de los corpus se realizó el mismo proceso de separación y guardado en los conjuntos de datos, cada uno con diferente variación en la cantidad de datos para entrenamiento y prueba, pero de la misma forma.

5.3 Experimentos

En cuanto a la experimentación se realizó desde la computadora descrita en el capítulo de integración de módulos en la sección 4.1. Se generaron diversos experimentos que son descritos a continuación, además de los diferentes resultados obtenidos que de igual forma son presentados a detalle en la sección 5.4.

5.3.1 Experimento con corpus español-inglés.

Para efectos de experimentación en el modelo generado para la traducción con la red profunda, se optó por realizar pruebas con el corpus español-inglés y así poder observar los diferentes escenarios que se pueden presentar al tener diversos volúmenes de datos para el entrenamiento y pruebas en un modelo de redes profundas.

Experimento 1.1

En esta experimentación se tomó en cuenta una parte de las oraciones que fueron seleccionadas aleatoriamente del corpus completo, el cual consta de 120,045 oraciones, donde las oraciones extraídas fueron guardadas en un elemento tabla de dos columnas con ayuda de la librería Pandas de Python.

La tabla 5.4 muestra un ejemplo de las oraciones tomadas en forma aleatoria con el número de la oración dentro del corpus, seguido de la oración en el idioma origen y la oración en el idioma destino.

Tabla 5.4 Visualización de las oraciones.

ID	eng	spa
7628	Do you know her?	¿Vos la conoces?
5253	Tom shot Mary.	Tom le disparó a Mary.
8222	I prefer biking.	Prefiero ir en bicicleta.
10430	He's watching me.	Me está observando.
9557	We're confident.	Estamos seguros.
6741	She's Canadian.	Es canadiense.
4202	I don't drink.	Yo no tomo.
7534	Can I borrow it?	¿Puedo pedirlo prestado?
11180	It was excessive.	Fue excesivo.
8049	I got sunburned.	Me tosté con el sol.

Dentro del modelo generado, para el entrenamiento de la red profunda, se consideró la oración origen como la entrada y para poder generar la salida por medio de una predicción hecha por la red, se tomó la oración destino como la etiqueta, de esta

manera el modelo realiza el entrenamiento tomando en consideración la etiqueta y así poder generar una oración destino.

La tabla 5.5 nos muestra la distribución del corpus español-inglés, que fue dividido al 10% del conjunto total de datos, con ayuda de la función de división de *sklearn*, lo que da un total de 12,025 oraciones para realizar el entrenamiento en la red profunda y la prueba del modelo.

Tabla 5.5 Distribución del corpus español-inglés en experimento 1.1.

		spa	eng
Entrenamiento (training)	Oraciones	8,417	
	Palabras	6,203	3,436
	Mayor longitud en oración	7	5
Prueba (test)	Oraciones	3,608	
	Palabras	3,556	2,277
	Mayor longitud en oración	7	5

En esta experimentación se consideraron 100 épocas de entrenamiento y un tamaño de lote de 64 muestras para el aprendizaje de la red profunda, además dentro de la validación que se tiene en el modelo se tomaron el 30% de los datos del conjunto asignado al entrenamiento, particionando el conjunto en 5,891 ejemplos para el entrenamiento y 2,526 ejemplos para la validación.

Experimento 1.2

La segunda experimentación se generó para verificar el funcionamiento del modelo siendo alimentado con un mayor número de oraciones, y poder así asentar varias teorías sobre el uso de las redes profundas.

Para efectos de esta experimentación, la distribución del conjunto de datos se muestra en la tabla 5.6, donde se tomaron aleatoriamente 90,000 del conjunto total de datos, lo cual representa un 75% de datos del corpus completo. Tomando en consideración que este incremento de datos aumentó el número de palabras que serán puestas a entrenar y la longitud de las oraciones de igual forma incrementó a casi el doble.

Tabla 5.6 Distribución del corpus español-inglés en experimento 1.2.

		spa	eng
Entrenamiento (training)	Oraciones	63,000	
	Palabras	27,842	16,024
	Mayor longitud en oración	15	13
Prueba (test)	Oraciones	27,000	
	Palabras	17,641	11,087
	Mayor longitud en oración	14	12

Además del incremento al conjunto de datos, también se modificaron varios parámetros para el entrenamiento de la red profunda, donde al tener mayor cantidad de datos es necesario modificar el lote de muestras, colocando a 128 el tamaño de las muestras que serán tomadas en el entrenamiento.

A esto se experimentó con el número de épocas, se consideraron 200 épocas pero se agregó el parámetro 'parada temprana' que nos ayuda a detener las iteraciones del entrenamiento cuando se obtiene el mejor parámetro designado y así guardar el modelo obtenido en esa iteración, que en este caso fue cuando el modelo encuentre la mejor 'exactitud' y en 10 iteraciones no supere el número, será detenido el entrenamiento quedando ese el mejor modelo obtenido.

5.3.2 Experimento con corpus español-LSE.

En esta experimentación se desarrollaron pruebas sobre el conjunto total de muestras que se tienen sobre el tema de lengua de señas, que a diferencia del corpus español-inglés, se tomaron en consideración diferentes parámetros de prueba para comprobar diferentes puntos de vista.

La tabla 5.7 muestra el conjunto total de datos que se usaron para realizar las experimentaciones, que contienen datos que son considerablemente menores a los ya probados con el corpus de la sección 5.3.2.

Tabla 5.7 Distribución del corpus español-LSE en experimentación.

		esp	LSE
Entrenamiento (training)	Oraciones	194	
	Palabras	890	627
	Mayor longitud en oración	34	40
Prueba (test)	Oraciones	35	
	Palabras	269	223
	Mayor longitud en oración	49	26

Este corpus contiene un total de 229 oraciones escritas de texto en español a texto en lengua de señas española, lo que representa uno de los mayores corpus que se tienen del tema.

En cuanto a la variación en la cantidad de datos que se usaron para entrenamiento y pruebas, se tomaron el 85% para el entrenamiento y el 15% para las pruebas, debido a que son pocas oraciones y se busca que el entrenamiento que pueda realizar la red profunda sea hecha con la mayor cantidad de información posible.

Experimento 2.1

La primera experimentación generada fue considerada en base a la segunda experimentación que se realizó con el corpus español-inglés, donde los puntos tomados en cuenta fueron el número de épocas para el entrenamiento, generando

1000 iteraciones para ajustar el modelo lo mejor posible, y reduciendo el tamaño del lote de muestras a 32, ya que el conjunto de datos es muy pequeño, y los parámetros deben de establecerse con la mayor precisión posible.

Al entrenar la red profunda se tomaron en cuenta 155 oraciones para hacer el entrenamiento y 39 oraciones para efectos de validación.

Uno de los puntos importantes en esta experimentación es el uso del parámetro 'parada temprana', que nos ayuda a detener el entrenamiento al momento de obtener el mejor valor en la 'exactitud' y no esperar a que termine de realizar las 1000 iteraciones para hacer uso del modelo.

Experimento 2.2

En la segunda experimentación se tomaron en cuenta la misma cantidad de épocas y el número de lote de muestras, por lo mencionado al tener pocos datos y ajustar los recursos del entrenamiento lo mejor posible.

A diferencia de la experimentación anterior, no se tomó en consideración el parámetro 'parada temprana', dejando el entrenamiento de la red profunda ser realizado con las 1000 iteraciones.

5.3.3 Experimento con corpus español-LSM.

Para efectos de experimentación con el corpus de español-LSM se utilizó el total del corpus en una sola corrida, además de generarse con 1000 épocas sin hacer uso del parámetro de 'parada temprana' y considerando el mínimo en el número de lote de muestras debido al bajo número en el conjunto de datos con el que se cuenta. Se usó el 90% del conjunto total de datos para el entrenamiento y

únicamente el 10% del conjunto de datos para efectos de pruebas debido al reducido número de muestras que se tiene, ejemplificado en la tabla 5.8.

Tabla 5.8 Distribución del corpus español-LSM en experimentación.

		esp	LSM
Entrenamiento (training)	Oraciones	9	
	Palabras	44	41
	Mayor longitud en oración	13	18
Prueba (test)	Oraciones	1	
	Palabras	9	13
	Mayor longitud en oración	9	8

5.3.4 Experimentos del traductor.

Para efectos de experimentación con el traductor, hablando del trabajo en conjunto de los módulos de procesamiento de lenguaje natural, análisis gramatical y traducción, se hizo uso de 10 oraciones generadas aleatoriamente, las cuales posteriormente se ingresaron al corpus LSM, que con ayuda de una persona Sorda fueron validadas las glosas y de esa manera se procedió a realizar la experimentación del traductor.

Cada una de las oraciones fue ingresada tanto al autómata como a la red profunda y así poder verificar, por medio de las métricas, cual es el valor que se está

obteniendo en la traducción para poder mostrar la mejor traducción obtenida hasta el momento en el presente trabajo.

5.4 Resultados

En este apartado se muestran los resultados que se obtuvieron al realizar las diferentes experimentaciones con cada uno de los corpus descritos en la sección 5.3, además de comparar y determinar cuáles son los mejores resultados en cuanto a desempeño y traducciones obtenidas.

5.4.1 Resultados con corpus español-inglés.

Los resultados que se obtuvieron en las experimentaciones realizadas con el primer corpus serán descritos a continuación de acuerdo a cada uno de los experimentos generados.

Resultados experimento 1.1

La experimentación realizada con el 10% del total del conjunto de datos del corpus que consta de 12,045 oraciones Como se puede observar en la tabla 5.9 se concluye con 100 iteraciones el experimento 1.1 obteniendo la mejor exactitud en la última iteración con un porcentaje del casi 54%.

Tabla 5.9 Resultados de las últimas tres épocas del experimento 1.1.

Época	Perdida	Exactitud	Perdida en Validación	Perdida en Exactitud
98	0.1931	0.5353	2.8066	0.2598
99	0.1873	0.5374	2.8133	0.2630
100	0.1784	0.5397	2.8219	0.2608

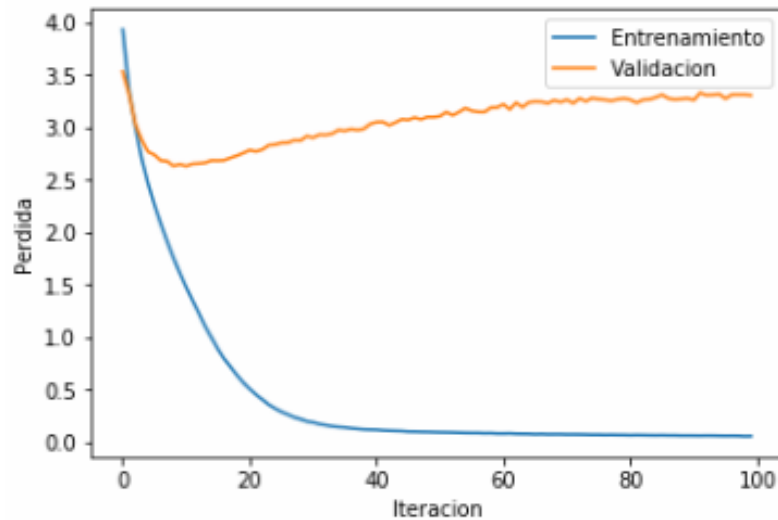


Figura 5.2 Comparación de pérdida en entrenamiento y validación del experimento 1.1.

Al momento en el que el modelo es entrenado, se realiza una validación cruzada con el 20% del conjunto de datos de prueba para poder llevar el valor de pérdida que se va obteniendo de la función de costo, esto se puede observar en la figura 5.2 donde se obtiene el mejor valor de pérdida en las primeras iteraciones y a partir de eso incrementa el valor de pérdida en el entrenamiento del modelo.

Los resultados de las oraciones resultantes son presentados en la tabla 5.10, en donde se puede observar la oración de entrada, la oración de salida correcta y la

oración de salida decodificada, que es resultado de la predicción que se obtuvo haciendo uso del modelo previamente entrenado.

Tabla 5.10 Oraciones predichas del experimento 1.1.

ID	Oración	Traducción Correcta	Predicción
3602	Olvídate de ella.	Forget about her.	Forget me.
2072	Puede que vengan.	They may come.	It may do.
2094	Tom encontró a Mary.	Tom found Mary.	Tom heard Mary.
1893	Lea después de mí.	Read after me.	Show after me.
3067	No hice nada.	I did nothing.	I didn't nothing.
2138	Yo puedo hacer ambos	I can do both.	I see on two.
1331	Tom escucha.	Tom listens.	Tom likes.
1150	Adelante.	Go ahead.	Go ahead.

Se puede observar que las predicciones tienen coherencia y cumplen con la estructura de lo que debería ser la oración correcta, aún le falta completar la oración correctamente pero la predicción de las oraciones se está realizando de manera correcta.

Resultados experimento 1.2

Los resultados que se obtuvieron del experimento 1.2 son mostrados a continuación, los cuales fueron generados con el 75% de los datos totales del corpus español-inglés y los cuales muestran un mejoramiento en las predicciones generadas por el modelo.

En la tabla 5.11 podemos observar los resultados que se generaron en las iteraciones del experimento 1.2, el cual se puso a iterar con 200 épocas pero se colocó un parámetro de 'parada temprana' para detener las iteraciones en el mejor modelo obtenido.

Tabla 5.11 Resultados en las iteraciones del experimento 1.2.

Época	Perdida	Perdida en Validación
1	3.7882	3.3150
2	3.0065	2.0309
3	2.6000	2.5918
4	2.3316	2.4579
198	0.1854	4.3413
199	0.1855	4.3368
200	0.1885	4.3078

El mejor modelo que se obtuvo fue generado en la iteración 9 con el mejor valor de pérdida sin embargo se generó otra prueba para dejar correr el experimento las 200 épocas y se puede observar que se obtiene el mismo valor en la iteración 9 y que al seguir iterando hasta la época 200 el valor de pérdida solamente tiende a incrementar.

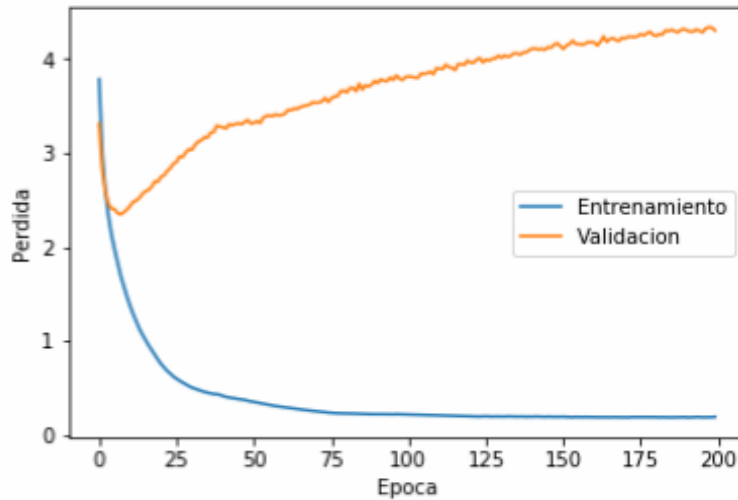


Figura 5.3 Comparación de pérdida en entrenamiento y validación del experimento 1.2.

La validación cruzada generada de la experimentación 1.2 nuevamente nos muestra que el mejor valor de pérdida se obtiene dentro de las primeras iteraciones y siguiente de esto comienza a incrementar la validación sobre el entrenamiento como se puede visualizar en la figura 5.3.

Al verificar los resultados de las oraciones del experimento 1.2, mostrados en la tabla 5.12, podemos determinar que el entrenamiento con mayores datos es más eficiente al obtener predicciones mejores, y al compararlos con las oraciones destino del corpus se puede observar que la oración tiene coherencia y es una buena traducción.

Tabla 5.12 Oraciones predichas del experimento 1.2.

ID	Oración	Traducción Correcta	Predicción
0	¿Te gustaría que llame a un taxi?	Would you like me to call a taxi?	Would you like to take call taxi?
1	Tom parecía muy triste.	Tom seemed very sad.	Tom looked very sad.
2	Será mejor que vayas a la cama.	You had better go to bed now.	You better go bed now.
3	No sé nada de él.	I don't know anything about him.	I know nothing anything about him.
4	Dame un poco de tiempo para pensar.	Give me some time to think.	Give me a little to think it.

Al comparar estas dos experimentaciones se observa un fenómeno curioso, primeramente el hecho de mejora en las predicciones obtenidas con un mayor número en el conjunto de datos que se le está proporcionando al modelo cuando realiza el entrenamiento que es algo predecible.

Lo segundo es que al dejar entrenar el modelo con el total de iteraciones asignadas, sin detenerlo con algún parámetro (en este caso las 200 iteraciones y sin usar la parada temprana), se obtiene un modelo que genera mejores predicciones en las traducciones que realiza. Esto es debido a que el modelo de embeddings, al tener un entrenamiento prolongado, genera un mejor aprendizaje de las palabras y combinaciones de palabras que va obteniendo, lo que lleva a generar una mejor predicción.

Pese a tener una exactitud por debajo de lo esperado, sigue presentando mejores resultados al realizar las predicciones de las traducciones.

5.4.2 Resultados con corpus español-LSE.

Las experimentaciones que se realizaron con el corpus de español-LSE fueron generadas con la misma cantidad en el conjunto de datos debido al poco contenido que existe sobre corpus de lengua de señas, y muestran los resultados siguientes:

Resultados experimento 2.1

Este experimento fue realizado con 229 oraciones, el conjunto total de oraciones del corpus que fue dividido en 194 oraciones de entrenamiento (85% del conjunto de datos) y 35 oraciones de prueba (15% de conjunto de datos) y los resultados en el entrenamiento se muestran en la tabla 5.13, donde podemos observar los resultados obtenidos de las iteraciones generadas en el experimento 2.1, el cual fue definido para realizar 1000 iteraciones pero se utilizó el parámetro de 'parada temprana' y obtiene el mejor modelo entrenado en la iteración 13.

Tabla 5.13 Resultados de las iteraciones del experimento 2.1.

Época	Perdida	Exactitud	Perdida en Validación	Perdida en Exactitud
1	5.8964	0.0655	5.7848	0.0948
2	5.6138	0.0830	5.4570	0.0948
3	5.3136	0.0825	5.3969	0.0948
4	5.2043	0.0835	5.4024	0.0948
5	5.1616	0.0840	5.4156	0.0948
6	5.1367	0.0825	5.4492	0.0948
7	5.1094	0.0838	5.4784	0.0948

8	5.0825	0.0881	5.4927	0.0968
9	5.0637	0.0886	5.5257	0.0968
10	5.0280	0.0909	5.5491	0.0968
11	5.0105	0.0916	5.5786	0.0968
12	4.9881	0.0911	5.6070	0.0968
13	4.9825	0.0909	5.6135	0.0968

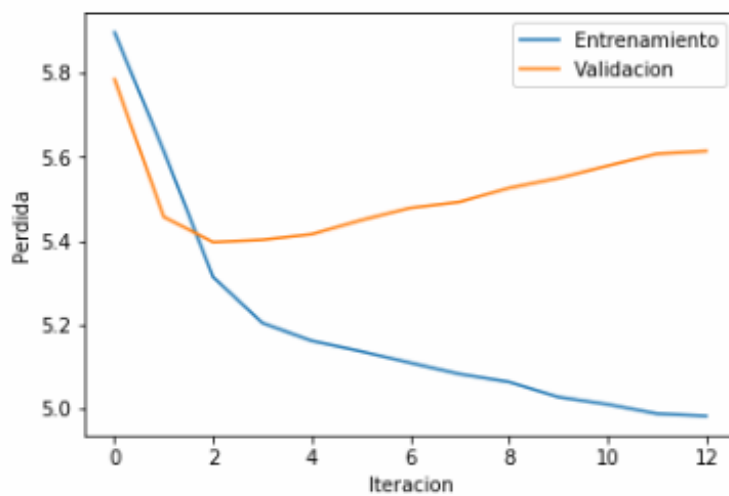


Figura 5.4 Comparación de pérdida en entrenamiento y validación del experimento 2.1.

La validación cruzada nos permite observar la variación que hay en la pérdida del entrenamiento y la validación a partir de las primeras iteraciones, la cual es mostrada en la figura 5.4, sin embargo con la parada temprana solamente llega a las 13 iteraciones.

En la tabla 5.14 se observan los resultados que se obtuvieron al usar el modelo entrenado y buscar obtener las predicciones, donde se visualiza que no logra generar ninguna predicción en ninguna de las oraciones que se le está ingresando al modelo.

Tabla 5.14 Oraciones obtenidas en experimento 2.1 del corpus esp-LSE.

ID	Oración	Traducción Correcta	Predicción
0	Los domingos papá compra churros.	DOMINGO PAPÁ CHURRO COMPRAR	Vacío
1	Al final se pone muy sucia de correr y tirarse por el suelo.	AL-FINAL, SUCIO MOTIVO CORRER SUELO SUCIO ROPA.	Vacío
2	También le gusta comer pan blanco.	TAMBIÉN GUSTAR QUÉ? PAN BLANCO	Vacío
3	Después de la comida mama siempre nos da un poco de chocolate.	DESPUÉS COMER FIN COMER, CHOCOLATE MAMÁ DAR-ME.	Vacío
4	Tras esto nos sentamos con papá y mamá en el salón.	FIN, SALÓN SENTAR PADRE-MADRE.	Vacío

Resultados experimento 2.2

La experimentación 2.2 fue realizada con los mismos parámetros que la experimentación 2.1, pero sin el uso del parámetro 'parada temprana' y nos muestra los siguientes resultados.

Los resultados de la tabla 5.15 muestran el entrenamiento del modelo con 1000 iteraciones señala que el modelo obtiene el mejor valor de pérdida en la iteración 9, sin embargo, al dejar el modelo entrenar sin hacer uso de la 'parada temprana' podemos observar que llega a obtenerse una exactitud cerca del 92% en este entrenamiento con 1000 iteraciones realizadas.

Tabla 5.15 Resultados en las iteraciones del experimento 2.2.

Época	Perdida	Exactitud	Perdida en Validación	Perdida en Exactitud
1	5.9096	0.0415	5.8382	0.0924
8	5.2491	0.0822	5.4487	0.0924
9	5.2208	0.0821	5.4415	0.0924
10	5.1969	0.0824	5.4423	0.0924
998	0.0295	0.9173	9.4858	0.1114
999	0.0294	0.9178	9.4894	0.1066
1000	0.0297	0.9179	9.5091	0.1114

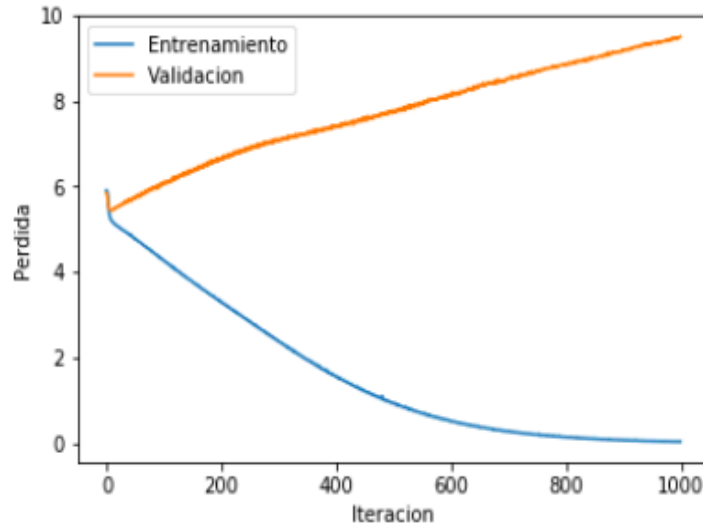


Figura 5.5 Comparación de pérdida en entrenamiento y validación del experimento 2.2.

En la figura 5.5 podemos observar la validación cruzada de la experimentación 2.2 que deja en claro la separación del valor de pérdida desde las primeras iteraciones y el comienzo del incremento en cada una de las iteraciones más agregadas al entrenamiento.

Las oraciones que se obtienen en la predicción con el modelo entrenado de este experimento resultaron ser positivas, ya que realiza una predicción que tiene coherencia y, pese a que no es acertado en comparación con las oraciones destino del corpus, se obtiene una oración estructurada con sentido como podemos observar en la tabla 5.16.

Tabla 5.16 Oraciones obtenidas en experimento 2.2 del corpus esp-LSE.

ID	Oración	Traducción Correcta	Predicción
0	Los domingos papá compra churros.	DOMINGO PAPÁ CHURRO COMPRAR	COMER PREPARAR
1	Primero suelto a mi perro y después le tiro la pelota.	PRIMERO YO CLI: “desatar al perro”, PELOTA TIRAR LEJOS,	PRIMERO PARO DINERO GRAN SUELDO NO.
2	Mientras que en España el peso de los hijos repercute más sobre la mujer no se podrá esperar un aumento de la fecundidad.	ENTONCES, ESPAÑA EN-ESTE-MOMENTO HIJO RESPONSABLE MÁS QUIÉN-? MUJER+, POSIBLE FUTURO MUJER NACIMIENTO++ CRECER-NO.	ESPAÑA ESPAÑA PRIMERO NACIMIENTO MUY POCO MAS O MENOS 115.
3	Ana siempre entra antes que yo.	MUJER (“Ana”) PRIMERO, YO SEGUNDO.	EL MUJER “ANA” Y MEDIA LOS DOS ESPECIAL GUERRA.
4	También le gusta comer pan blanco.	TAMBIÉN GUSTAR QUÉ? PAN BLANCO.	PAPÁ CUANTO LEER ESCRIBIR MUY INTELIGENTE.

En comparación con el experimento 2.1, el realizar el entrenamiento y finalizarlo sin ‘parada temprana’ presentó una mejoría tanto en la exactitud obtenida del modelo, como en las predicciones, sin embargo, pese a este hecho, no cumple en su totalidad con una oración bien traducida ya que no tiene acercamiento a la oración destino que resulta de la oración origen que se está decodificando, lo que nos lleva a deducir que lo analizado en el experimento 1.1, la cantidad de oraciones del

corpus con el que se esté realizando el entrenamiento del modelo si influye al momento de obtener buenas predicciones.

5.4.3 Resultados con corpus español-LSM.

Los resultados que se obtuvieron al realizar la experimentación 3.1 con el corpus LSM, pese a ser mínima la cantidad de datos, fueron resultados prometedores y se muestran en la tabla 5.17.

Tabla 5.17 Resultado de las iteraciones del experimento 3.1.

Época	Perdida	Exactitud	Perdida en Validación	Perdida en Exactitud
1	3.1203	0.0000	2.9688	0.1000
2	3.0992	0.1824	2.9662	0.2000
3	3.0774	0.1834	2.9631	0.2000
24	2.3348	0.2020	2.8444	0.2000
25	2.3026	0.2039	2.8534	0.2000
26	2.2772	0.2028	2.8412	0.2000
27	2.2539	0.2039	2.8544	0.2000
998	1.3962e-07	0.8407	7.1928	0.2000
999	1.3229e-07	0.8353	7.1934	0.2000
1000	1.4206e-07	0.8394	7.1939	0.2000

Se muestra el avance que va teniendo en el proceso de entrenamiento con todas las iteraciones, donde el mejor modelo generado por la red profunda se presenta en

la iteración 26, sin embargo al dejar correr el total de iteraciones asignadas se genera un modelo con una exactitud del 83%, que a comparación del modelo de LSE en la experimentación 2.2 no es mejor, pero por la cantidad de datos en comparación de los dos corpus es totalmente ventajoso el resultado del modelo de LSE contra el modelo de LSM.

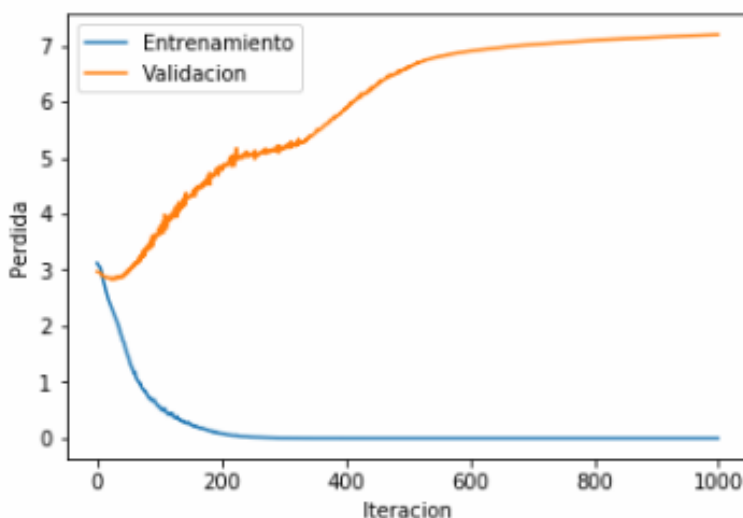


Figura 5.6 Comparación de pérdida en entrenamiento y validación del experimento 3.1.

La figura 5.6 muestra la validación cruzada en el entrenamiento con el corpus LSM, quien como todos los experimentos anteriores muestra la mejor pérdida en la validación en las primeras iteraciones a comparación del entrenamiento que se mantiene en una pérdida constante.

El resultado que se observa en la tabla 5.18 se generó de la predicción realizada con el modelo entrenado del corpus LSM, la cual no es precisa en su traducción pero se obtiene una oración con sentido. Más adelante se observará la comparación con los resultados obtenidos por el modelo LSM y el de LSE.

Tabla 5.18 Oración obtenida en el experimento 3.1 con el corpus LSM.

ID	Oración	Traducción Correcta	Predicción
0	Mañana tengo que ir al doctor con mi papá.	MAÑANA + PAPÁ + MÍO + YO + DOCTOR + IR.	MAÑAÑANA + DOS + PLATANOS + UNA + PERA + LUIS + COMER.

5.4.4 Resultados del traductor.

Al realizar la integración de cada uno de los módulos y comenzar con la experimentación del trabajo del traductor en conjunto mostró los resultados que podemos observar en la tabla 5.19, donde la traducción con el autómata muestra mayor coherencia y estructura que la mostrada por la traducción resultante de la red profunda.

Tabla 5.19 Oraciones resultado del autómata y red profunda.

Texto Original	Traducción Autómata Glosa LSM	Traducción Red Profunda Glosa LSM
Mi tío baila en la fiesta	FIESTA + TÍO MÍO + BAILAR.	LA HIJOS SEVILLA.

Comenzando por la oración origen que se trabaja en el módulo de procesamiento de lenguaje natural donde se obtienen los lemas de cada palabra identificada, y con la cual se procede a realizar una limpieza al texto lematizado para así pasar la

oración limpia, tanto al autómata como al modelo generado por la red profunda, y de esa forma obtener los resultados en el módulo de traducción

```
1 score_bleu_a, score_gleu_a, score_wer_a  
(0.7071067811865475, 0.4, 0.25)  
  
1 score_bleu_n, score_gleu_n, score_wer_n  
(0, 0.0, 1.0)
```

Figura 5.7 Evaluación de las traducciones con las métricas.

En la figura 5.7 se muestran los resultados en las evaluaciones realizadas por las métricas y aplicadas a cada una de las traducciones que se obtienen del autómata y la red profunda. La comparación realizada se obtuvo con la oración en glosa de la prueba manual tomada como referencia y al utilizar cada una de las traducciones como posibles hipótesis, se compararon y se obtuvieron los resultados mostrados, donde se puede determinar que la mejor traducción obtenida es la del autómata con una exactitud de 0.70 y un error del 0.25 en la traducción, lo que por el contrario de la traducción obtenida con la red profunda es de una exactitud de 0.0 y un error de 1.00, lo que significa que la traducción no fue correcta en ningún sentido.

Con esto el traductor puede generar una salida validada automáticamente y así poder asegurar que es la mejor traducción obtenida hasta el momento por el traductor.

5.4.5 Comparación de los resultados.

A continuación se pueden revisar los resultados obtenidos con cada uno de los modelos generados en cada una de sus experimentaciones, midiendo los resultados obtenidos con las métricas utilizadas dentro del traductor para verificar un promedio de la exactitud con la que se están realizando las traducciones y el porcentaje de error que está teniendo cada uno de los modelos, todo registrado en la tabla 5.20.

Tabla 5.20 Comparación de resultados.

Modelo	Corpus	Experimento	Exactitud	BLEU	WER
Red Profunda	spa-eng	E1.1	0.5397	0.51	0.66
		E1.2	0.4934	<u>0.4238</u>	<u>0.375</u>
	esp-LSE	E2.1	0.0909	0.0	1.0
		E2.2	0.9179	0.0	1.0
	esp-LSM	E3.1	<u>0.8394</u>	0.5946	1.0
Autómata	Usando reglas	E4.1	S/M	0.7071	0.25
		E4.2	S/M	0.8801	0.8
		E4.3	S/M	0.5623	0.6

Como se puede observar, el mejor resultado que se obtuvo en cuanto a la exactitud se consiguió en el experimento 2.2 con el corpus español-LSE, sin embargo las traducciones resultantes fueron catalogadas por las métricas como traducciones erróneas. Para poder visualizar la comparación al usar un modelo con datos

abundantes, como lo es el corpus español-inglés, con un modelo con datos mínimos, se puede observar la evaluación resultante de las métricas en el experimento 1.2, que fue la mejor catalogada de los modelos usando redes profundas por las métricas con las mejores traducciones obtenidas, sin tomar en cuenta las traducciones resultantes del modelo usando autómeta.

Por otra parte, las traducciones que mejor puntuación obtuvieron fueron las realizadas por el modelo del autómeta, que aunque no tuvieron medición de exactitud (sin medición o S/M) si fueron evaluadas por las métricas obteniendo los mejores resultados con las traducciones mejor elaboradas.

CAPÍTULO 6: Conclusiones y trabajo futuro

*“Siempre parece imposible hasta que se
hace”.*

– Nelson Mandela.

En este capítulo se presentan las conclusiones que se obtuvieron de la investigación descrita en esta tesis, las cuales parten de los objetivos que se plantearon al inicio del trabajo y son mostradas según los resultados que se obtuvieron en el estudio que se realizó.

Además de eso, se propone el trabajo futuro considerado para elaborarse y poder aplicar mejoras en lo que ya fue desarrollado en este trabajo.

6.1 Conclusiones

En términos generales, en el desarrollo de este proyecto se consiguió implementar un traductor que permite obtener un texto en glosa de LSM a partir de cualquier texto en español que se le ingrese. El desarrollo del traductor a su vez permitió implementar y comparar los métodos tradicionales de traducción de texto con el uso del enfoque de aprendizaje profundo en la tarea de traducción, consiguiendo así significativos resultados que ayudarán en un futuro para su aplicación en el área de procesamiento de texto.

La implementación del traductor permitió trabajar el área de procesamiento de lenguaje natural, a la que se le atribuye gran parte del trabajo que se genera para poder llegar al objetivo de obtener un texto en glosa. El estudio que se le dedicó a este módulo es clave en lo que se consiguió.

Por otra parte, el estado del arte nos mostró las técnicas aplicadas para la traducción del texto y de esta forma se tomó la decisión de aplicarlas y al mismo tiempo comparar los métodos tradicionales, que en este caso fue la implementación de un autómata, con la actual tecnología de inteligencia artificial ocupada en el área de traducción que ha conseguido resultados muy buenos en poco tiempo, la

implementación de una red profunda, que de acuerdo a la literatura se optó por hacer uso de la variación de una red recurrente que ayudó en la generación de un modelo capaz de predecir la glosa de LSM a partir del aprendizaje de oraciones que forman parte de un corpus.

Los resultados que se mostraron nos ayudaron a dejar el registro de un corpus de LSM, que es parte importante de los resultados generados por parte de la red profunda, el cual se comprobó que al alimentar la red profunda de un conjunto de datos de gran número, obtiene muy buenos resultados.

Un punto importante a recalcar sobre el módulo de traducción desarrollado es su utilidad, ya que fue realizado con la finalidad de integrarlo a la plataforma para la asistencia de personas con discapacidad auditiva y de esta forma poder generar parte de una herramienta altamente efectiva para el apoyo visual a las personas Sordas.

Además, este proyecto es útil para abrir un nicho en la investigación y desarrollo de proyectos para la comunidad de Sordos existentes en el país, ya que es un número grande de personas que pertenecen a la comunidad y que no reciben el apoyo correspondiente. Así con esto es posible seguir trabajando en desarrollos tecnológico que puedan ayudar con la problemática a la que se enfrentan y poder lograr la inclusión en cada uno de los aspectos de comunicación en las que somos partícipes en la vida cotidiana.

6.2 Trabajo futuro

Al inicio de la investigación se optó por delimitar el trabajo a estructuras textuales que forman parte de la estructura perteneciente a la lengua de señas, lo que nos lleva a plantear como posibilidad de trabajo a futuro la implementación de un módulo que trabaje los movimientos corporales y, de igual forma, otro módulo que trabaje los gestos faciales, logrando de esta manera poder abarcar la estructura completa de la lengua de señas.

Al observar los resultados que se obtuvieron en la presente investigación, hablando particularmente de la implementación del modelo de aprendizaje profundo, se propone continuar con la generación de oraciones para seguir alimentando el corpus de LSM, y en un futuro, al obtener un mayor número de oraciones en el conjunto de datos, se pueda volver a repetir la generación del modelo y observar los resultados que presente en las nuevas traducciones.

Bibliografía

- [1] J. Lobera, "Discapacidad auditiva Guía didáctica para la inclusión en educación inicial y básica," *Cons. Nac. Fom. Educ.*, p. 71, 2010.
- [2] W. M. Center, "Organizacion Mundial de la Salud," 2017. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs300/es/>. [Accessed: 21-Oct-2017].
- [3] INEGI, "INEGI. Instituto Nacional de Estadística y Geografía.," *Censos y Conteos de Población y Vivienda*, 2010. [Online]. Available: <http://www.beta.inegi.org.mx/temas/discapacidad/default.html>. [Accessed: 21-Oct-2017].
- [4] S. Arana, "De 12 mil sordos, sólo 7 estudian la universidad," *El Imparcial*, Hermosillo, pp. 1–3, 29-Nov-2015.
- [5] J. Esparza, "Uniradio Noticias," *Circulo Sonora*, 2016. [Online]. Available: <http://uniradionoticias.com/noticias/reportajesespeciales/432786/carece-sistema-educativo-en-sonora-de-interpretres-de-lenguaje-de-senas.html>. [Accessed: 25-Oct-2017].
- [6] CONADIS, "gob.mx," *Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad*, 2016. [Online]. Available: <https://www.gob.mx/conadis/articulos/lengua-de-senas-mexicana-lsm?idiom=es>. [Accessed: 24-Oct-2017].
- [7] M. C. Aldrete, *Gramática de la lengua de señas mexicana*. México, D.F.: Centro de Estudios Lingüísticos y Literarios, 2008.
- [8] R. San-Segundo, R. Barra, L. D'Haro, J. Montero, R. Córdoba, and J. Ferreiros, "A Spanish Speech to Sign Language Translation System for assisting deaf-mute people," *Ninth Int. Conf. Spok. Lang. Process.*, pp. 1399–1402, 2006.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," *Nat. Methods*, vol. 13, no. 1, pp. 35–35, 2015.
- [10] E. Brill and R. J. Mooney, "Overview of empirical natural language processing," *AI Mag.*, vol. 18, no. 4, pp. 13–24, 1997.
- [11] Microsoft, "Microsoft Cognitive Services," *Microsoft Azure*, 2017. [Online]. Available: <https://azure.microsoft.com/en-us/services/cognitive-services/>. [Accessed: 06-Apr-2018].
- [12] J. M. S. Corzo, D. D. Portillo, and J. A. C. Zamora, "Sistema de consultas en lenguaje natural para bases de datos," *CEUR Workshop Proc.*, vol. 220, 2006.
- [13] E. Rich, K. Knight, and S. Nair, "Artificial Intelligence (Third Edition)." p. 640, 2008.
- [14] Robert Corbett, "BISON 3.0.4: Bison Parser," *GNU Bison*, 2015. [Online].

Available: http://www.gnu.org/software/bison/manual/html_node/Bison-Parser.html.

- [15] Hanspeter Mössenböck, “COCO/R for C#,” *Computer Science System Software*, 2018. [Online]. Available: <http://www.ssw.uni-linz.ac.at/Coco/>. [Accessed: 10-May-2018].
- [16] R. Díez, “¿Qué es la búsqueda semántica y qué herramientas usar?,” *Locco*, 2018. [Online]. Available: <https://www.loccocommittee.com/busqueda-semantica-google/>. [Accessed: 10-May-2018].
- [17] Propellernet Ltd, “ANSWER THE PUBLIC,” *AnswerThePublic*, 2018. [Online]. Available: <https://answerthepublic.com/>.
- [18] Leow Kah Thong, “Keyword Tool,” *Key Tools Limited.*, 2018. [Online]. Available: <https://keywordtool.io/google>. [Accessed: 10-May-2018].
- [19] A. Fernández-Montraveta, G. Vázquez, and C. Fellbaum, “The Spanish version of WordNet 3.0,” *Text Resour. Lex. Knowl. 9th Conf. Nat. Lang. Process. KONVENS 2008*, vol. 33, pp. 175–182, 2008.
- [20] J. Verschueren, “The Pragmatic Return to Meaning: Notes on the Dynamics of Communication, Degrees of Saliency, and Communicative Transparency,” *J. Linguist. Anthropology*, vol. 5, no. 2, pp. 127–157, 1995.
- [21] I. Ivern, “La pragmática: un marco teórico para interpretar la comunicación,” *Rev. Logop. Foniatr. y Audiol.*, vol. 38, no. 1, pp. 1–5, 2018.
- [22] M. I. Mikeleiz, “Representación conexionista y procesamiento del discurso,” *An. Psicol.*, vol. 15, no. 2, pp. 303–318, 1999.
- [23] S. Sun, C. Luo, and J. Chen, “A review of natural language processing techniques for opinion mining systems,” *Inf. Fusion*, vol. 36, pp. 10–25, 2017.
- [24] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, vol. 43. 2009.
- [25] Apache, “Apache OpenNLP Developer Documentation,” *The Apache Software Foundation*, 2017. [Online]. Available: <http://opennlp.apache.org/docs/1.8.4/manual/opennlp.html>.
- [26] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr.*, pp. 55–60, 2014.
- [27] L. (Upc) Padro, “Analizadores Multilingües en FreeLing,” *Dept. Lenguajes y Sist. Informáticos Cent. Investig. TALP Univ. Politéc. Catalunya*.
- [28] Alexander Gelbukh y Grigori Sidorov, *CIENCIA DE LA COMPUTACIÓN: Procesamiento automático del español con enfoque en recursos léxicos grandes*. 2006.
- [29] A. Hurtado Albir, *Traducción y Traductología*. 2001.

- [30] Manuel Maldonado Alemán, *Aspectos teórico-prácticos de la traducción*, Graficas C. Madrid, 1989.
- [31] Real Academia Española, “Diccionario de la Real Academia Española,” 2018. [Online]. Available: <http://www.rae.es/obras-academicas/diccionarios/diccionario-de-la-lengua-espanola>. [Accessed: 12-Apr-2018].
- [32] T. P. Szynalski, “Collins COBUILD Advanced Dictionary,” *HarperCollins*, 1987. [Online]. Available: <http://www.antimoon.com/how/cobuild-review.htm>. [Accessed: 12-Apr-2018].
- [33] M. Kellogg, “WordReference.com LLC,” *Digital Millennium Copyright Act*, 1999. [Online]. Available: <http://www.wordreference.com/english/AboutUs.aspx>. [Accessed: 12-Apr-2018].
- [34] B. Turovsky, “Found in translation: More accurate, fluent sentences in Google Translate,” *Product Lead, Google Translate*, 2016. [Online]. Available: <https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/>. [Accessed: 12-Apr-2018].
- [35] Microsoft, “Microsoft Translator Text API,” *Bing Speech API*, 2018. [Online]. Available: <https://www.microsoft.com/en-us/translator/translatorapi.aspx>. [Accessed: 12-Apr-2018].
- [36] Svetlana Sokolova, “PROMT,” *PROMT Translation Software and Dictionaries*, 1991. [Online]. Available: <http://www.promt.com/>. [Accessed: 12-Apr-2018].
- [37] Tecnológico de Monterrey, “Procesador de palabras WORD,” 1997. [Online]. Available: http://www.cca.org.mx/cca/cursos/hbi-webtec/modulos/modulo5/aula/word_revision_tutor.htm. [Accessed: 21-Apr-2018].
- [38] Apache Software Foundation, “Apache OpenOffice 4.1.5 released,” *OpenOffice*, 2012. [Online]. Available: <https://www.openoffice.org/support/books.html>. [Accessed: 12-Apr-2018].
- [39] O. Pichardo-lagunas and B. Martinez-seis, “Resource Creation for Automatic Translation System from Texts in Spanish into Mexican Sign Language,” vol. 100, pp. 129–137, 2015.
- [40] M. Á. Rodríguez González, “Lenguaje de signos,” *Biblioteca Virtual Miguel de Cervantes*, 2003. [Online]. Available: <http://www.cervantesvirtual.com/nd/ark:/59851/bmc08633>. [Accessed: 10-May-2018].
- [41] E. R. Moreno, “Signslator, el primer traductor automático para el lenguaje de signos,” *Revistadigital*, 2016. [Online]. Available: <https://revistadigital.inesem.es/idiomas/signslator-traductor-lenguaje-de-signos/>. [Accessed: 10-Apr-2018].

- [42] Textosign, “Software to convert text into Spanish Sign Language,” *Fundacion Vodafone España*, 2011. [Online]. Available: <http://www.textosign.es/index.php/en/textosign-ique-es-textosign>. [Accessed: 10-May-2018].
- [43] D. G. Quinto-Pozos, “Contact between Mexican Sign Language (LSM) and American Sign Language (ASL) in two Texas border areas,” vol. 2, 2004.
- [44] B. Fridman, “Categorías verbales de aspecto y tiempo en la Lengua de señas mexicana,” in *Dimensiones del aspecto en español*, 2005, pp. 189–236.
- [45] D. J. M. Hortensia García y Lanz, Gloria Arango Mejía, *Mis primeras señas: una introducción al lenguaje manual*. Secretaría de Educación Pública, 1983.
- [46] A. Gonz, “Aprendizaje profundo para el procesamiento del lenguaje natural,” 2017.
- [47] G. Seif, “I’ll tell you why Deep Learning is so popular and in demand,” *The Startup*, 2018. [Online]. Available: <https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780>. [Accessed: 24-May-2018].
- [48] L. Deng and D. Yu, “Deep Learning: Methods and Applications,” *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197--387, 2013.
- [49] D. J. Matich, “Redes Neuronales: Conceptos Básicos y Aplicaciones.,” *Historia Santiago.*, p. 55, 2001.
- [50] S. Raschka, *Python Machine Learning*, no. 1. 2014.
- [51] The Jupyter Trademark, “Jupyter,” *Project Jupyter*, 2014. [Online]. Available: <http://jupyter.org/documentation>. [Accessed: 11-May-2018].
- [52] J. Hunter, D. Darren, F. Eric, and M. Droettboom, “matplotlib,” *Matlib development*, 2012. [Online]. Available: <https://matplotlib.org/>. [Accessed: 11-May-2018].
- [53] T. Augspurger, C. Bartak, P. Cloud, and A. Hayden, “Pandas,” *NumFOCUS*, 2018. [Online]. Available: <http://pandas.pydata.org/pandas-docs/stable/>. [Accessed: 11-May-2018].
- [54] F. Pedregosa, J. Grobler, and O. Grisel, “Scikit-Learn,” *INRIA*, 2010. [Online]. Available: <http://scikit-learn.org/stable/about.html#funding>. [Accessed: 11-May-2018].
- [55] T. Oliphant, P. Peterson, and E. Jones, “SciPy,” *SciPy developers*, 2001. [Online]. Available: <https://www.scipy.org/about.html>. [Accessed: 11-May-2018].
- [56] M. Abadi *et al.*, “TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning,” *12th USENIX Symp. Oper. Syst. Des. Implement. (OSDI ’16)*, pp. 265–284, 2016.

- [57] A. Unruh, "What is the TensorFlow machine intelligence platform?," *Opensource.com*, 2017. [Online]. Available: <https://opensource.com/article/17/11/intro-tensorflow>. [Accessed: 11-May-2018].
- [58] F. Chollet, "Keras," *MIT*, 2015. [Online]. Available: <https://keras.io/>. [Accessed: 11-May-2018].
- [59] S. Baldassarri and F. J. R. Santos, "Traductor de Español a LSE 1 basado en reglas gramaticales y morfológicas," *España*, pp. 13–22, 2006.
- [60] S. Baldassarri, E. Cerezo, and F. Royo-Santas, "Automatic translation system to Spanish sign language with a virtual interpreter," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5726 LNCS, no. PART 1, pp. 196–199, 2009.
- [61] V. López-Ludeña, C. González-Morcillo, J. C. López, R. Barra-Chicote, R. Cordoba, and R. San-Segundo, "Translating bus information into sign language for deaf people," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 258–269, 2014.
- [62] M. Serafin and R. González, *Manos con Voz Diccionario de Lengua de Señas Mexicana*. 2011.
- [63] L. A. García López, R. Rodríguez Cervantes, M. Zamora Martínez, and S. S. Esteban Sosa, "Mis Manos Que Hablan. Lenguaje De Señas Para Sordos." p. 326, 2006.
- [64] O. Pichardo-Lagunas, L. Partida-Terrón, B. Martínez-Seis, A. Alvear-Gallegos, and R. Serrano-Olea, "Sistema de traducción directa de español a LSM con reglas marcadas," *Adv. Nat. Lang. Process. Comput. Linguist.*, vol. 115, pp. 29–42, 2016.
- [65] O. Caraveo-Carvajal, A. L. Millán-Castro, B. Cota Ponce, M. T. Serna-Encinas, and C. E. Rose-Gómez, "Arquitectura Propuesta de Reconocimiento de Voz para Traducción a Lengua de Señas Mexicana Mediante un Avatar," *Av. Investig. en Ing. en el Estado Son.*, vol. 3, no. 1, pp. 371–376, 2017.
- [66] E. D. Barraza-Granillo, C. E. Rose-Gómez, S. Gonzáles-López, and M. T. Serna-Encinas, A. L. Millán-Castro, "Arquitectura Propuesta de un Sistema de Apoyo Educativo para Personas con Discapacidad Auditiva," *Av. Investig. en Ing. en el Estado Son.*, vol. 3, no. 1, pp. 225–232, 2017.
- [67] J. M. Mager Hois, "Traductor hibrido wixarika-español con escasos recursos bilingües", Ciudad de Mexico., pp. 64-84, 2017.
- [68] Ch. Kelly, "Tab-delimited bilingual sentence pairs" *Tatoeba Project*, 2019. [Online]. Available: <http://www.manythings.org/anki/>. [Accessed: 23-Nov-2018].
- [69] J. Porta Zamorano, "Towards a rule-based spanish to spanish sign language translation: from written forms to phonological representations", Madrid, pp. 85-116, 2014.

- [70] O. E. Caraveo Carvajal, "Sistema de reconocimiento de voz y traducción a lenguaje de señas mexicano mediante un avatar", Tesis de maestría, Tecnológico Nacional de México/ I.T. Hermosillo, Hermosillo, Son., MX, 2018
- [71] E. D. Barraza Granillo, "Plataforma para la asistencia de personas con discapacidad auditiva en el contexto de la educación", Tesis de maestría, Tecnológico Nacional de México/ I.T. Hermosillo, Hermosillo, Son., MX, 2018
- [72] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311-318, 2002.