



Universidad de las Ciencias  
Informáticas

**Facultad 1**

# **Analizador semántico para un sistema de comunicación aumentativa y alternativa pictotraductor**

Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:**

Bento de Assunção Cassule Cussei

**Tutores:**

PAs. Laritza González Marrero

PAs. Dargel Veloz Morales

La Habana, 2022

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Bento de Assunção Cassule Cussei  
(Autor)

---

PAs. Laritza González Marrero  
(Tutor)

---

PAs. Dargel Veloz Morales  
(Tutor)

## RESUMEN

Los Sistemas de Comunicación Aumentativos y Alternativos (SCAA) son herramientas importantes para sujetos con trastornos de comunicación (por e.j, personas con trastornos de espectro autista, síndrome de Down y parálisis cerebral). Los avances actuales en el campo de las TICs han permitido un importante impulso en el desarrollo de esos sistemas. En este trabajo se presenta Pictobana, un analizador semántico para un sistema de comunicación aumentativa y alternativa pictotraductor. El modelo fue desarrollado aplicando técnicas de procesamiento del lenguaje natural. Se realiza un análisis lingüístico con el objetivo de proporcionar las mejores representaciones posibles de los textos en pictogramas. El modelo puede ser usado en una aplicación web, este proporciona una herramienta que ayuda a promover las competencias y habilidades de comunicación a personas con dificultades del habla en Cuba y a sus familiares. Las pruebas realizadas mediante experimentos y criterio de expertos, demuestran que el analizador desarrollado, aumenta la estabilidad de los pictogramas al contexto y a la semántica, aminorando la incoherencia y la ambigüedad semántica del futuro sistema.

**Palabras clave:** Comunicación Aumentativa y Alternativa, SCAA, Pictogramas, Pictotraductor, Procesamiento del Lenguaje Natural, Semántica, Pictobana.

## LISTADO DE FIGURAS

Figura 1. Traducción de la oración “mamá está en la cocina”.....	25
Figura 2. Traducción de la oración “mamá está en la cocina”.....	25
Figura 3. Secuencia del proceso CRISP-DM.....	26
Figura 4. Características de los pictogramas de la API de ARASAAC .....	44
Figura 5. Ejemplo de dataframe de pandas.....	44
Figura 6. Ejemplo de pictogramas con el mismo nombre (cocina) .....	45
Figura 7. Ejemplo de etiquetado gramatical.....	46
Figura 8. Los 5 primeros resultados de la búsqueda para la cadena “el” usando los embeddings de los nombres.....	47
Figura 9. Los 5 primeros resultados de la búsqueda para la cadena “el” usando los embeddings de las descripciones.....	47
Figura 10. Los 5 primeros resultados de la búsqueda para la cadena “papá” usando los embeddings de las descripciones.....	47
Figura 11. Los 5 primeros resultados de la búsqueda para la cadena “papá” usando los embeddings de los nombres. ....	48
Figura 12. Etiquetado gramatical de la oración “Yo prefiero el vuelo de la mañana para la habana”. ....	49
Figura 13. Los 5 primeros resultados de la búsqueda para la palabra “prefiero” usando los embeddings de los nombres.....	49
Figura 14. Los 5 primeros resultados de la búsqueda para la palabra “preferir” usando los embeddings de los nombres.....	50
Figura 15. Traducción de la oración “ <i>Yo prefiero el vuelo de la mañana</i> ” sin hacer fragmentación.....	51
Figura 16. Traducción de la oración “ <i>Yo prefiero el vuelo de la mañana</i> ” teniendo en cuenta la fragmentación.....	51
Figura 17. Representación de Palabras en un Espacio Vectorial.....	55
Figura 18. Analogía entre palabras.....	56

Figura 19. La relación de palabras en una oración que plantea la necesidad de incrustaciones compuestas.....	57
Figura 20. Alimentando oraciones con palabras faltantes al modelo BERT para entrenamiento.....	60
Figura 21. Arquitectura del codificador cruzado BERT.....	61
Figura 22. Arquitectura SBERT para calcular puntuaciones de similitud.....	62
Figura 23. Embeddings de los nombres. Resultado de la codificación y creación de los embeddings de los nombres.....	64
Figura 24. Embeddings de las descripciones. Resultado de la codificación y creación de los embeddings de las descripciones.....	64
Figura 25. Arquitectura del analizador semántico.....	65
Figura 26. Los dos primeros resultados al traducir la oración “Cierra el grifo al terminar”.....	70
Figura 27. Resultados, porcentaje de acierto en la primera posición.....	71
Figura 28. Resultados, porcentaje de acierto en las cinco primeras posiciones.....	71

## LISTADO DE TABLAS

Tabla 1. Operacionalización de las variables.....	13
Tabla 2. Análisis de los sistemas homólogos.....	24
Tabla 3. Formas flexionadas y sus lemas/raíz.....	48
Tabla 4. Datos de los expertos seleccionados.....	68
Tabla 5. Resultados de la evaluación de Pictobana en comparación a Pictar y Pictotraductor.....	72

## LISTA DE ABREVIACIONES Y SIGLAS

<b>ARASAAC</b>	Portal Aragonés para la Comunicación Aumentativa y Alternativa
<b>AVC</b>	Accidente Vascular Cerebral
<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>CAA</b>	Comunicación Aumentativa e Alternativa
<b>Crisp-DM</b>	Cross-Industry Standard Process for Data Mining
<b>HNSWLIB</b>	Hierarchical Navigable Small Worlds Library
<b>NCR</b>	National Cash Register
<b>PLN</b>	Procesamiento del Lenguaje Natural
<b>POS</b>	Part Of Speech
<b>RAE</b>	Real Academia Española
<b>REN</b>	Reconocimiento de Entidades Nombradas
<b>SBERT</b>	Sentence Bidirectional Encoder Representations from Transformers
<b>SCAA</b>	Sistema Comunicación Aumentativa e Alternativa
<b>TEA</b>	Trastornos del Espectro Autista

# ÍNDICE

<b>INTRODUCCIÓN</b>	10
<b>CAPÍTULO 1: DEL PROCESAMIENTO DEL LENGUAJE NATURAL, HERRAMIENTAS Y METODOLOGÍA</b>	16
1.1 Conceptos Fundamentales	16
Procesamiento del lenguaje natural	16
Semántica	16
Contexto	17
Pictograma	18
La Comunicación Aumentativa y Alternativa	18
Sistema de Comunicación Aumentativa y Alternativa	19
Pictotraductor	21
1.2 Sistemas Homólogos (sistemas/características)	22
Pictotraductor	22
AraTraductor	22
Pictar	23
1.3 Metodología	25
CRISP-DM	25
1.4 Herramientas y Tecnologías	35
ARASAAC	35
Python 3.7.14	35
Anaconda 2022.05	36
Google Colab	36
Spacy 3.4.0	37
Pandas 1.3.5	38
HNSWLIB	38
SentenceTransformers	38
Lucidchart	38



1.5 Conclusiones Parciales	39
<b>CAPÍTULO 2: ANALIZADOR SEMÁNTICO: DISEÑO, CONSTRUCCIÓN Y METODOLOGÍA</b>	40
2.1 Comprensión del negocio/problema	40
2.2 Análisis exploratorio de la base de datos ARASAAC	42
2.3 Análisis Lingüístico	44
Tokenización	45
Etiquetado Gramatical	45
Lematización	47
Fragmentación (Chunking)	49
2.4 Modelado	51
2.4.1 Búsqueda Semántica	51
2.4.2 BERT	52
Vectorización (Word Embeddings)	52
Transformador (Transformer)	55
¿Qué es BERT?	57
Sentence BERT	60
2.4.3 Codificación y Creación de los Embeddings de los Nombres y de las Descripciones	61
Embeddings de los Nombres	62
Embeddings de las descripciones	62
2.5 Arquitectura del modelo creado	62
2.6 Evaluación de la Solución Propuesta	65
2.7 Conclusiones Parciales	71
<b>CONCLUSIONES</b>	72
<b>RECOMENDACIONES</b>	73
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	74

## INTRODUCCIÓN

La comunicación aumentativa y alternativa (CAA) es un área de la práctica clínica que complementa o compensa las deficiencias en la producción y/o comprensión del habla y el lenguaje, incluidos los modos de comunicación hablados y escritos. CAA cae bajo el paraguas más amplio de tecnología de asistencia, o el uso de cualquier equipo, herramienta o estrategia para mejorar la vida diaria funcional en personas con discapacidades o limitaciones (ASHA 2022a). La comunicación es el proceso activo de intercambiar información e ideas. La comunicación implica tanto la comprensión como la expresión (ASHA 2022b). El objetivo de la CAA es lograr la comunicación más efectiva para el individuo con el fin de maximizar su potencial, independencia e inclusión social, para llevar la mejor calidad de vida posible (AAC INSTITUTE 2022).

La CAA utiliza una variedad de técnicas y herramientas tecnológicas y/o no tecnológicas para ayudar al individuo a expresar sus pensamientos, deseos y necesidades, sentimientos e ideas (ASHA 2022a). Una de esas técnicas es el uso de tableros de comunicación con imágenes o pictogramas que representan un objeto, persona, lugar o concepto. Los sistemas informáticos construidos con esta técnica son ampliamente llamados "Pictotraductores", puesto que su función principal es traducir texto a pictogramas. En este contexto es de suma importancia que las frases construidas con esos sistemas sean lingüísticamente correctas. Para lograr una traducción/construcción que no comprometa el aprendizaje de personas con dificultad del habla, "estos sistemas deben aplicar una variedad de estrategias para la construcción de oraciones que sean sintáctica y semánticamente correctas"(Franco et al. 2018).

El análisis semántico (Maulud et al. 2021) es una característica esencial del enfoque de procesamiento del lenguaje natural (PLN). Indica, en el formato apropiado, el contexto de una oración o párrafo. La semántica trata sobre el estudio de la significación del lenguaje. El análisis semántico en estos sistemas significa entregar el pictograma correcto según el significado de cada palabra o expresión dependiendo del contexto del texto de entrada, lo que le hace crucial para la obtención de mejores resultados en las traducciones (FRANCO 2020).

La aparición de las innovaciones tecnológicas en el último medio siglo ha tenido un enorme impacto en las posibilidades de aprendizaje (Toukoumidis, Rodríguez, Rodríguez 2018). Actualmente existen investigaciones y pictotraductores para fines diversos como por ejemplo, la gestión dinámica de agendas [(*teapps* 2022); (Zorrilla Villanueva 2018); (*Pictoagenda* sin fecha)], leer noticias en pictogramas (*PictoNews :: Creando-y-educando* 2018), construcción de oraciones mediante predicciones de pictogramas (Pereira et al. 2022) y pictotraductores con fines de traducir oraciones a pictogramas [(*Pictotraductor* 2013); (Martín et al. 2018); (Bautista et al. 2017)]. No obstante, algunos no siempre son eficientes en sus traducciones, otros no son de código abierto, los más sofisticados como SCAA integrados en tabletas tienen un alto costo económico, y debido a la escasez de recursos en los países latinos e hispanohablantes, la mayoría de las investigaciones que presentan resultados prometedores son realizadas y/o probadas en el idioma inglés, limitando el acceso a un determinado público.

En Cuba existen esfuerzos y metodologías de enseñanza para personas con dificultad del habla, sin embargo estas metodologías usan libros, libretas y/o cartulina (donde son representados los pictogramas) como materiales de estudios; por el momento aún no hay un sistema de CAA de alta tecnología para auxiliar en la enseñanza de este grupo de personas. Según (Bautista et al. 2017) esta forma de comunicación ha cobrado mayor relevancia en relación con las nuevas tecnologías, esto significa que muchas personas con discapacidad del habla corren un alto riesgo de exclusión social ya que se encuentran sin acceso a formas clave de comunicación como la escritura y la lectura.

Pese a todos estos logros y esfuerzos realizados en este ámbito, aún se hace notar la necesidad de correcciones semánticas para los picto-traductores existentes, sobre todo en la región hispana, dando lugar a las siguientes situaciones problemáticas:

Existen conflictos semánticos al procesar palabras que se escriben igual, pero poseen diferentes significados, permitiendo la inserción de respuestas incoherentes y mezcla de contextos no deseada.

Persisten respuestas semánticamente distantes al contexto en que se desarrolla la comunicación, como consecuencia se introducen ambigüedades que producen confusión a los que poseen dificultades de lenguaje.

Por lo anteriormente planteado se define como **problema de investigación**: ¿Cómo desarrollar un analizador semántico para aminorar la ambigüedad e incoherencia semántica en las traducciones de los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor?

Se enmarca el **objeto de estudio** en analizadores semánticos en los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor.

Se determina como **campo de acción**: Los analizadores semánticos para los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor en Cuba.

Se plantea como **objetivo general**: Desarrollar un analizador semántico para aminorar la ambigüedad e incoherencia semántica en las traducciones de los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor.

Por consiguiente los **objetivos específicos** son los siguientes:

1. Caracterizar los fundamentos teórico-metodológicos que permitan el desarrollo de la solución propuesta sobre el analizador semántico para los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor.
2. Recopilar un dataset para crear el analizador semántico.
3. Implementar de la propuesta del analizador semántico.
4. Validar el analizador semántico a fin de probar su efectividad para la traducción de texto a pictogramas.

Teniendo en cuenta el problema y el objetivo planteado, se define como **hipótesis** que el desarrollo de un analizador semántico aminora la ambigüedad y la incoherencia semántica en las traducciones de los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor.

Tabla 1. Operacionalización de las variables.

Variable conceptual	Dimensión	Indicadores	Sub-indicadores	Unidad de Medida
<b>Analizador Semántico</b>	Pictotraductor	Funcionalidad	Funcionalidad	(%)
		Fragmentación	Tokenización	(%)
			Agrupamiento	(%)
<b>Incoherencia semántica en los sistemas de comunicación aumentativa</b>	Personas con necesidades especiales de comunicación	Ajustabilidad de los pictogramas al contexto	Se ajusta	0.5
			No se ajusta	0
<b>Ambigüedad Semántica en los sistemas de comunicación aumentativa</b>	Personas con necesidades especiales de comunicación	Ajustabilidad de los Pictogramas a la semántica	Se ajusta	0.5
			No se ajusta	0

Para cumplir el objetivo propuesto se requieren las siguientes **tareas investigativas**:

1. Delimitación del problema y caracterización del estado de:
  - a. Sistemas de Comunicación Aumentativa y Alternativa
  - b. Pictotraductores
  - c. Analizadores semánticos
  - d. Las tendencias en la construcción de analizadores semánticos
2. Recopilación del dataset ideal para crear el analizador semántico.
3. Implementación de la propuesta de solución.
4. Validación de la propuesta de solución.

El alcance de este trabajo es limitado a 2 aspectos:

1. Idioma - este trabajo es desarrollado para la lengua española.

2. Público alvo - este trabajo es desarrollado para personas con dificultad del habla sin problemas visuales.

Los métodos científicos que se utilizan en la presente investigación son:

#### **Métodos teóricos:**

Histórico-Lógico: para estudiar la trayectoria y el desarrollo histórico de los analizadores semánticos hasta las tendencias actuales.

Analítico-sintético: para analizar la documentación recopilada, seleccionando los elementos esenciales para darle solución al problema planteado.

Modelación: para modelar el proceso de desarrollo de la investigación mediante los diagramas correspondientes a la fase de análisis y diseño, proporcionando una mejor perspectiva de la aplicación a desarrollar.

#### **Métodos empíricos:**

Observación: para conocer el comportamiento actual de los sistemas de CAA.

Análisis documental: para consultar la literatura especializada en temáticas afines a la investigación.

Entrevista: para entrevistas a especialistas del área de lingüística y que trabajan con niños con dificultad del habla.

A continuación, se describe brevemente la estructura del presente trabajo de diploma:

### **Capítulo 1: Del procesamiento del lenguaje natural, herramientas y metodología**

Se realiza un análisis de los principales conceptos asociados al Procesamiento del Lenguaje Natural y Sistemas de Comunicación Aumentativa y Alternativa que son necesarios para la comprensión de la presente investigación. Además, se describen las herramientas y la metodología para el desarrollo de la solución.

### **Capítulo 2: Analizador semántico: diseño, construcción y pruebas**

Se describe la propuesta de solución, los detalles relacionados con la implementación de la propuesta de solución, materiales y métodos utilizados en la construcción del analizador semántico. Se especifican los escenarios de pruebas realizados al

analizador semántico para comprobar que cumple con los requerimientos establecidos, y se presentan los resultados del mismo.

## **CAPÍTULO 1: DEL PROCESAMIENTO DEL LENGUAJE NATURAL, HERRAMIENTAS Y METODOLOGÍA**

En el presente capítulo se realiza un análisis de los principales conceptos asociados al Procesamiento del Lenguaje Natural y Sistemas de Comunicación Aumentativa y Alternativa, necesarios para la comprensión de la presente investigación. Se estudian dichos sistemas y se proponen la metodología, lenguaje y herramientas para la construcción del analizador semántico.

### **1.1 Conceptos Fundamentales**

Con el objetivo de comprender la presente investigación, a continuación se presentan los referentes teóricos que sirven de base para esta pesquisa.

#### **Procesamiento del lenguaje natural**

Por Procesamiento de Lenguaje Natural (PLN, denominado también NLP por sus siglas en inglés) se entiende la habilidad de la máquina para procesar la información comunicada, no simplemente las letras o los sonidos del lenguaje. La ciencia que estudia el PLN se llama lingüística computacional. El nombre fue inventado en los tiempos cuando eso era: lingüística para las computadoras. Los lingüistas, a través de la introspección e intuición, escribían las reglas y los diccionarios cada vez más exactos y detallados, acercándose al objetivo: dotar a la computadora con la capacidad de entender el lenguaje humano (Gelbukh 2010).

#### **Semántica**

Según (Espinal et al. 2020), en la Lingüística contemporánea es habitual distinguir dos disciplinas dedicadas al estudio del significado. La Semántica trata sobre el significado lingüístico de las palabras y el significado resultante de su combinación. La Pragmática permite analizar diferentes aspectos del significado implícito y los procesos inferenciales que regulan la comunicación (Félix-Brasdefer 2018). Esta división metodológica entre disciplinas que tratan del estudio del significado responde a una concepción teórica de acuerdo con la cual, la primera se basa fundamentalmente en parámetros lingüísticos y estructurales, mientras que la segunda tiene en cuenta cómo interaccionan parámetros lingüísticos con parámetros



no lingüísticos y contextuales. La primera persigue explicar el significado lingüístico, mientras que la segunda persigue explicar la interpretación de enunciados.

Partiendo de la oración de ejemplo (1):

Deberías aceptarlos. (1)

Según (Espinal et al. 2020), si se quiere explicar el significado de (1) deben tenerse en cuenta varios elementos. En primer lugar, el significado de los elementos léxicos. La Semántica da cuenta del hecho de que deber es un verbo que puede tener un significado epistémico “es necesario” o un significado deóntico “es obligatorio”. También da cuenta de la ambigüedad de aceptar, ya que este verbo significa, entre otras acepciones, “Recibir o dar entrada”, pero también “Asumir resignadamente un sacrificio, molestia o privación” (RAE 2022).

### **Contexto**

Según (Mandujano et al. 2021) comprender el significado de una palabra durante la lectura implica reconocer la identidad y la posición de las letras dentro de la palabra. Este proceso de decodificación parece más acertado cuando la palabra se encuentra dentro de un contexto semántico que la antecede. La noción de contexto proviene del latín *contextus* que significa conexión, coherencia o una acción de poner un todo en conjunto. La (RAE 2022) lo define de la siguiente manera: el “entorno lingüístico del que depende el sentido de una palabra, frase o fragmento determinados,” “el entorno físico o de situación, político, histórico, cultural o de cualquier otra índole en el que se considera un hecho” o “la unión de cosas que se enlazan y entretienen.” El contexto es un concepto fundamental en la pragmática que ha sido debatido e interpretado desde distintas perspectivas en la lingüística, la psicología, la sociología, la antropología lingüística, el discurso, y el discurso mediado por computadoras, entre otros.

Aunque se han propuesto diferentes tipos de contextos, en esta investigación interesa uno: el contexto lingüístico:

El **contexto lingüístico**, también llamado **cotexto**, según (Félix-Brasdefer 2018) se refiere a la información lingüística que precede o sigue a un enunciado. Por ejemplo, Pedro: ¿Quién llegó tarde a la clase? Sonia: Luis. Lo que dijo Pedro en el discurso

anterior juega un papel importante para entender la construcción elíptica usada por Sonia (Luis [llegó tarde a clase]).

## **Pictograma**

La (RAE 2022) define el pictograma como “signo de la escritura de figuras o símbolos”. Los pictogramas fueron diseñados por Roxana Mayer Johnson en 1981; y se utilizan como sistemas alternativos a la comunicación oral (Barranquero, Baeza 2012). Es la representación del lenguaje mediante objetos reales, fotos, dibujos o pictogramas. En algunos casos se coloca con la palabra correspondiente en la parte superior y en cartulinas de diferentes colores (Nakamura, Zeng-Treitler 2012). El método de los pictogramas es uno de los programas más efectivos que se usan en la enseñanza estructurada, ofrece a las personas con autismo las guías que necesitan para saber qué es lo que van a hacer, cómo, con quién y cuánto tiempo va a estar en esa actividad (Acuña, Mendoza 2020).

## **La Comunicación Aumentativa y Alternativa**

La CAA (ASHA 2022a) es un área de la práctica clínica que complementa o compensa las deficiencias en la producción y/o comprensión del habla y el lenguaje, incluidos los modos de comunicación hablados y escritos. CAA cae bajo el paraguas más amplio de tecnología de asistencia, o el uso de cualquier equipo, herramienta o estrategia para mejorar la vida diaria funcional en personas con discapacidades o limitaciones (p.ej., personas con Parálisis Cerebral, Microcefalia, Trastornos del Espectro Autista (TEA), secuelas de Accidentes Vasculares Cerebrales (AVC) y Apraxia).

La comunicación aumentativa y alternativa (AAC) incorpora una amplia gama de procesos que aumentan, complementan o reemplazan el habla de las personas con necesidades de comunicación complejas. En el amplio contexto del habla y el lenguaje, el habla a menudo se asocia con los movimientos motores responsables de la producción de palabras habladas, mientras que el lenguaje se asocia con las habilidades de procesamiento cognitivo de la comunicación (Elsahar et al. 2019).

Las soluciones de CAA se clasifican en tres categorías: CAA sin tecnología, de baja tecnología y de alta tecnología (Cook, Polgar 2014). La CAA sin tecnología se

considera la más antigua de las tres categorías de CAA, dada su dependencia de la interpretación de las expresiones faciales y los movimientos motores voluntarios, como el lenguaje de señas, para transmitir mensajes no verbales. La CAA de baja tecnología utiliza herramientas básicas, como libros y tableros de anuncios con léxicos extensos de imágenes y frases para ayudar en el proceso de comunicación. La CAA de alta tecnología abarca el uso de dispositivos electrónicos para lograr un objetivo de CAA. Los dispositivos que se incluyen en esta categoría, como los dispositivos inteligentes y los dispositivos AAC dedicados, integran hardware y software para satisfacer las necesidades de comunicación de un usuario. Un atributo común de las soluciones de CAA modernas tiende a basarse en la traducción de los significados previstos del usuario al habla a través de dispositivos generadores de voz. La comunicación CAA también se clasifica a menudo como sin ayuda o con ayuda, dada la dependencia de la solución del cuerpo humano únicamente o la interacción con una ayuda comunicativa externa para la comunicación, respectivamente (Elsahar et al. 2019).

Generalmente, los sistemas CAA de alta tecnología están equipados con imágenes (p.ej., fotos y pictogramas), que representan palabras o expresiones, a menudo utilizadas en la construcción de frases telegráficas (es decir, que se basan únicamente en palabras clave, sin el uso de conjunciones, preposiciones, artículos, conjugación de verbos, etc.) (PEREIRA 2020).

### **Sistema de Comunicación Aumentativa y Alternativa**

Según (Marín Martínez 2019) la explicación más completa de los sistemas de Comunicación Aumentativa y Alternativa (SCAA), la aporta Tamariz (1988), que los define como “instrumentos de intervención logopédico -educativos, destinados a personas con alteraciones diversas de la comunicación y/o del lenguaje, cuyo objetivo es la enseñanza mediante procedimientos específicos de instrucción, de un conjunto estructurado de códigos no vocales, necesitados o no de soporte físico, sirven para llevar a cabo actos de comunicación funcional espontánea y generalizable”.

La comunicación aumentativa y alternativa (CAA) abarca todo tipo de métodos de comunicación que complementan y facilitan, o incluso reemplazan, el lenguaje hablado y/o escrito (Cabello et al. 2018).

La principal diferencia que existe entre los sistemas alternativos y aumentativos de comunicación es que los sistemas alternativos sustituyen al lenguaje oral, mientras que los aumentativos actúan como ayuda o complemento del mismo. De esta manera, se habla de comunicación alternativa cuando se refiere a cualquier forma de comunicación distinta del habla y que se emplea en situaciones de comunicación cara a cara mientras que la comunicación aumentativa se refiere a la comunicación de apoyo o ayuda, que se utiliza con un doble objetivo: promover y apoyar el habla y garantizar la comunicación cuando el habla por sí misma no es suficiente (Marín Martínez 2019).

Es por ello que las personas que pueden beneficiarse del uso de los SCAA son aquellas que presentan necesidades especiales de comunicación, de forma permanente o transitoria. Según (Marín Martínez 2019), los usuarios se pueden clasificar en:

- Personas con diversidad funcional motora.
- Personas con diversidad funcional intelectual.
- Personas con diversidad funcional sensorial.
- Personas con necesidades transitorias.

Los sistemas alternativos de comunicación persiguen distintas finalidades, las cuales dependen de las características individuales de cada usuario. Según (Marín Martínez 2019), dichas finalidades pretenden :

1. Servir como medio de comunicación alternativo para aquellas personas que carecen de lenguaje oral.
2. Servir como medio de comunicación aumentativa para aquellas personas que presentan un lenguaje oral muy limitado.
3. Servir como medio de comunicación transitorio para aquellas personas que, a pesar de tener posibilidades de comunicación oral, esta ha sido interrumpida por algún suceso temporal.

Para considerar un SCAA como un sistema de comunicación, es necesario que cumpla unos requisitos. Estos requisitos son los siguientes tal y como señala (Marín Martínez 2019):

- Cubrir toda la escala de funciones de la comunicación.
- Ser compatible con otros aspectos de la vida del individuo.
- Facilitar la comunicación con cualquier interlocutor, que pueda ser utilizado en distintos entornos teniendo las menos restricciones posibles.
- Ser efectivo, extensible y adaptable al desarrollo y evolución de las capacidades y posibilidades del sujeto.
- Tener un uso motivador y fácil, a la par que asequible.

Los Sistema de Comunicación Aumentativa y Alternativa se pueden clasificar según (Tönsing et al. 2019), en dos tipos de sistemas, sin apoyo y con apoyo:

- Sistemas sin apoyo: son aquellos que no precisan de ningún elemento físico externo al emisor. Es el propio emisor, quien a través de su propio cuerpo (gestos y signos manuales) configura y transmite el mensaje. Dentro de estos, se encuentran el lenguaje de señas, gestos, el sistema bimodal y la palabra complementada.
- Sistemas con apoyo: requieren de algún tipo de apoyo o soporte físico para su utilización, en este caso se usan elementos tangibles y símbolos gráficos (fotografías, pictogramas...), además de los soportes necesarios, los cuales pueden tratarse desde sencillos soportes como cartulinas, tableros de madera o plástico, hasta tecnologías más sofisticadas, como las Tablet y las computadoras. Dentro de estos, se encuentran los pictogramas, elementos no estructurados (por e.j, uso de fotografías) y diferentes softwares.

### **Pictotraductor**

Según (Rodríguez Velecela, Juela León 2021) es un sistema de comunicación aumentativa y alternativa de alta tecnología que permite traducir oraciones a pictogramas.

## 1.2 Sistemas homólogos (sistemas/características)

En esta sección se presentan los sistemas homólogos. En las siguientes secciones se busca responder las siguientes preguntas:

**P1.** ¿Cómo se utiliza el análisis semántico en el sistema?

**P2.** ¿Dónde fue desarrollado?

**P3.** ¿El sistema toma en cuenta el contexto de las oraciones?

**P4.** ¿Cuál es el nivel de accesibilidad del sistema?

### **Pictotraductor**

Pictotraductor (*Pictotraductor* 2013) es un traductor de texto a pictogramas en forma de aplicación web totalmente gratuita desarrollada por el Grupo-Promedia (*Grupo-Promedia* sin fecha). Está pensada para intentar facilitar la comunicación, con personas que tienen dificultades de expresión mediante el lenguaje oral y que se comunican más eficientemente mediante imágenes a través de pictogramas cuya base es la colección ARASAAC (elCorreoGallego 2013). Pictotraductor recibe como entrada texto plano en español, y devuelve una traducción en pictogramas que consiste prácticamente en una traducción palabra por palabra (Martín et al. 2018).

### **AraTraductor**

AraTraductor es un traductor de texto español a pictogramas que se apoya en técnicas de Procesamiento del Lenguaje Natural (PLN) para mejorar la traducción final obtenida. El objetivo de la aplicación es ayudar a los usuarios que no pueden leer el lenguaje natural pero entienden los pictogramas para que puedan traducir automáticamente los textos escritos a sus equivalentes de pictogramas. Estos usuarios incluyen tanto usuarios finales con discapacidad como tutores y familias que quieren elaborar materiales en pictogramas para sus alumnos o hermanos (Bautista et al. 2017). La traducción de texto a pictogramas del Pictotraductor consta de cuatro pasos diferentes: análisis sintáctico, preprocesamiento, consulta a la base de datos y procesamiento de n-gramas.

El sistema empieza por realizar un análisis sintáctico del texto utilizando la biblioteca Maltparser (Nivre et al. 2007). Para cada palabra, se centran en su POS (sustantivo, adjetivo, signo de puntuación o determinante, por ejemplo) y su lema (la palabra base que corresponde a los plurales o tiempos verbales). Con la información obtenida del análisis sintáctico, se procesa el texto para eliminar la información que no se utiliza en la traducción. Por ejemplo, los signos de puntuación como comas o puntos se eliminan de las oraciones. Además, las palabras como determinantes también se eliminan del texto. Para cada palabra del texto preprocesado, el sistema realiza una serie de consultas a la base de datos de pictogramas. Se busca la palabra original y si se encuentra se almacena el pictograma correspondiente para la traducción. Si no se encuentra la palabra original, se dispara una segunda consulta utilizando el lema de la palabra obtenido en el análisis sintáctico. El resultado de este paso es una lista de pictogramas que corresponden a las palabras que aún quedan en el texto. Por último, el sistema realiza un procesamiento a partir de cada palabra para ver si junto a las siguientes forma un n-grama con pictograma asociado en ARASAAC.

## **Pictar**

PICTAR es un traductor de texto a pictogramas en forma de aplicación web que tiene como objetivo facilitar el trabajo con pictogramas en el ámbito educativo, desde tareas como la edición de materiales por parte de los docentes, como su uso por parte de los alumnos en las actividades dentro del aula cuando sea posible c.

Una vez se recibe el texto el sistema realiza primeramente un análisis morfológico del mismo para obtener las categorías gramaticales y lemas de las palabras. La importancia de conseguir el lema de las palabras reside en cómo se organiza la base de datos de ARASAAC, ya que por lo general cada pictograma está relacionado sólo con el lema de una palabra. Por ejemplo, existe un pictograma para el verbo “comer” pero no para ninguna de sus derivaciones, o para el adjetivo “feliz” pero no para “felices”. Una vez realizado el análisis, para cada palabra se almacena su lema y su categoría gramatical. En lugar de realizar una simple correspondencia palabra-picto, se realiza un tratamiento adicional de posibles n-gramas (sucesiones de n palabras cuya unión puede tener un significado conjunto y estar representada por uno o más pictogramas). Así, el servicio realiza un procesamiento a partir de cada palabra para ver si junto a las siguientes forman un n-grama con pictograma asociado en

ARASAAC. El resultado final del servicio de traducción contiene las palabras del texto (agrupadas como n-gramas donde corresponda).

Tabla 2. Análisis de los sistemas homólogos

Sistema	Uso de análisis semántico	Contexto	Accesibilidad	Lugar de desarrollo
<b>Pictotraductor</b>	No	No	Alta	España
<b>AraTraductor</b>	No	-	-	España
<b>Pictar</b>	No	No	Alta	España

Todos los sistemas analizados presentan limitaciones en las preguntas planteadas. Los tres sistemas fueron desarrollados en España, aunque sea un país hispano hablante su cultura no necesariamente aborda un lenguaje que pueda ser comprendido por una persona con dificultades del lenguaje en Cuba. Algunas terminologías difieren del lenguaje cotidiano de un cubano, lo que puede dificultar su comprensión cuando son traducidas en pictogramas. También se pudo observar que ninguno realiza un análisis semántico; sin embargo el AraTraductor y el Pictar se destacan por hacer un buen análisis sintáctico morfológico (técnica de procesamiento del lenguaje natural) que es usado para mejorar los resultados. Pictar, presenta cierta deficiencia en las representaciones de los n-gramas como destaca en su documentación. En cambio, el AraTraductor no fue posible tener acceso al sistema, solamente a la documentación. Por último, el Pictotraductor hace una traducción palabra por palabra a pictograma. Ninguno tiene en cuenta el contexto y solamente el Pictotraductor y Pictar están abiertos al público.



Figura 1. Traducción de la oración "mamá está en la cocina" (Pictotraductor 2013)



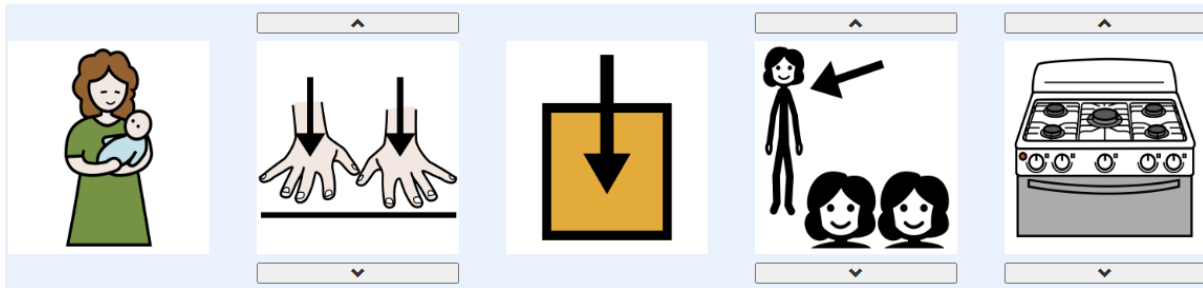


Figura 2. Traducción de la oración "mamá está en la cocina" (PICTAR sin fecha)

Luego de este análisis se pudo observar que, aunque algunos hacen algún tipo de procesamiento de texto, como se puede observar en las figuras 1 y 2 ninguno resuelve el problema de ambigüedad semántica en los sistemas de CAA. En adición, estos sistemas fueron desarrollados

A continuación, se muestran la metodología, herramientas y tecnologías que contribuyen a darle solución a este problema.

### 1.3 Metodología

#### CRISP-DM

Crisp-DM (IBM 2021) (Cross-Industry Standard Process for Data Mining en inglés y Proceso estándar entre industrias para la minería de datos en español) es una organización europea creada por tres grandes jugadores en proyectos de minería de datos que son SPSS (Soluciones de productos y servicios estadísticos, en español), NCR Corporation (Caja Registradora Nacional, en español) y Daimler Chrysler. Esta metodología trata de desarrollar los proyectos de minería de datos bajo un proceso estandarizado de definición y validación de tal forma que se desarrollen proyectos minimizando los costos implicados y con un alto impacto en el negocio.

Según (Contreras Arteaga, Sánchez Cotrina 2020) la metodología CRISP-DM proporciona dos documentos como herramienta de ayuda en el desarrollo del proyecto de minería de datos: el modelo de referencia y la guía del usuario. El documento modelo de referencia describe de forma general las fases, tareas generales y salidas de un proyecto de minería en general. La guía del usuario proporciona información más detallada sobre la aplicación práctica del modelo de referencia a proyectos de minería de datos específicos, proporcionando consejos y

listas de comprobación sobre las tareas correspondientes a cada fase. La metodología CRISP-DM estructura el ciclo de vida de un proyecto de minería de datos en seis fases, que interactúan entre ellas de forma iterativa durante el desarrollo del proyecto.

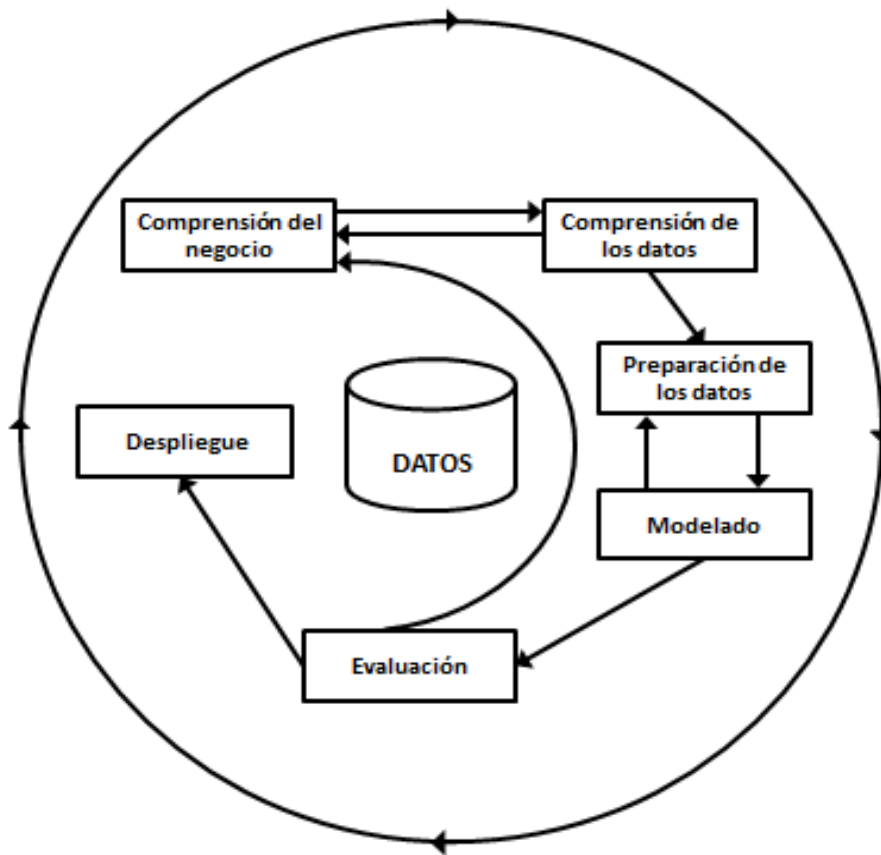


Figura 3. Secuencia del proceso CRISP-DM (IBM 2021)

En la Figura 3, las flechas indican las dependencias más importantes y frecuentes entre fases. La secuencia de las fases no es estricta. De hecho, la mayoría de los proyectos avanzan y retroceden entre las fases según sea necesario. El círculo exterior simboliza la naturaleza cíclica del proceso de modelado.

CRISP-DM es una forma probada en la industria de guiar los proyectos de minería de datos. Como metodología, incluye descripciones de las fases típicas de un proyecto, las tareas involucradas en cada fase y una explicación de las relaciones entre estas tareas. Como modelo de proceso, CRISP-DM proporciona una descripción general del ciclo de vida de la minería de datos (Figura 3).

A continuación se explica cada una de estas fases (Rodríguez 2010):

## **1. Comprensión del negocio**

Esta primera fase es probablemente la más importante y aglutina las tareas de comprensión de los objetivos y requisitos del proyecto desde una perspectiva de negocio, con el fin de convertirlos en objetivos técnicos y en un plan de proyecto. Sin lograr comprender dichos objetivos, ningún algoritmo por muy sofisticado que sea, permitirá obtener resultados fiables. Para obtener el mejor provecho de la minería de datos, es necesario entender de la manera más completa el problema que se desea resolver, esto permitirá recolectar los datos correctos e interpretar correctamente los resultados. En esta fase, es muy importante la capacidad de poder convertir el conocimiento adquirido del negocio en un problema de minería de datos y en un plan preliminar cuya meta sea el alcanzar los objetivos del negocio. A continuación se muestra una descripción de cada una de las principales tareas que componen esta fase.

### **a. Determinar los objetivos del negocio**

Esta es la primera tarea a desarrollar y tiene como metas determinar cuál es el problema que se desea resolver, por qué la necesidad de utilizar la minería de datos y definir los criterios de éxito. Los problemas pueden ser diversos, como por ejemplo, detectar fraude en el uso de tarjetas de crédito, detección de intentos de ingreso indebido a un sistema, asegurar el éxito de una determinada campaña publicitaria, etc. En cuanto a los criterios de éxito, estos pueden ser de tipo cualitativo, en cuyo caso un experto en el área de dominio califica el resultado del proceso de minería de datos, o bien de tipo cuantitativo.

### **b. Evaluación de la situación**

En esta tarea se debe calificar el estado de la situación antes de iniciar el proceso de minería de datos, considerando aspectos tales como: ¿cuál es el conocimiento previo disponible acerca del problema?, ¿se cuenta con la cantidad de datos requerida para resolver el problema?, ¿cuál es la relación coste beneficio de la aplicación de minería de datos?, etc. En esta fase se definen los requisitos del problema, tanto en términos de negocio como en términos de minería de datos.

**c. Determinar los objetivos de la minería de datos**

Esta tarea tiene como objetivo representar los objetivos del negocio en términos de las metas del proyecto de minería de datos.

**d. Realizar el plan del proyecto**

Esta última tarea de la primera fase de CRISP-DM tiene como meta desarrollar un plan para el proyecto, que describa los pasos a seguir y las técnicas a emplear en cada uno de ellos.

**2. Comprensión de los datos**

Esta segunda fase comprende la recolección inicial de los datos con el objetivo de establecer un primer contacto con el problema, familiarizarse con ellos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis. Esta fase junto a las dos siguientes fases son las que demandan el mayor esfuerzo y tiempo en un proyecto de minería de datos.

A continuación, se muestran las tareas que componen esta fase.

**a. Recolectar los datos iniciales**

La primera tarea en esta segunda fase del proceso de CRISP-DM es la recolección de los datos iniciales y su adecuación para el futuro procesamiento. Esta tarea tiene como objetivo elaborar informes con una lista de los datos adquiridos, su localización, las técnicas utilizadas en su recolección y los problemas y soluciones inherentes a este proceso.

**b. Descripción de los datos**

Después de adquiridos los datos iniciales, estos deben ser descritos. Este proceso implica establecer volúmenes de datos (número de registros y campos por registro), su identificación, el significado de cada campo y la descripción del formato inicial.

**c. Exploración de los datos**

Una vez realizada la descripción de los datos, se procede a su exploración, cuyo fin es encontrar una estructura general para los datos. Esto implica la aplicación de pruebas estadísticas básicas que revelen propiedades en los datos recién adquiridos, se crean tablas de frecuencia y se construyen gráficos de distribución. La salida de esta tarea es un informe de exploración de los datos.

#### **d. Verificar la calidad de los datos**

En esta tarea se efectúan verificaciones sobre los datos para determinar la consistencia de los valores individuales de los campos, la cantidad y distribución de los valores nulos, y para encontrar valores fuera de rango, los cuales pueden constituirse en ruido para el proceso. La idea, una vez llegados a este punto, es poder garantizar la completitud y corrección de los datos.

### **3. Preparación de los datos**

En esta fase y una vez efectuada la recolección inicial de los datos, se procede a su preparación para adaptarlos a las técnicas de minería de datos que se van a utilizar posteriormente, éstas pueden ser técnicas de visualización de datos, de búsqueda de relaciones entre variables u otras medidas para explotación de los datos. La preparación de los datos incluye las tareas generales de selección de datos a los que se va a aplicar una determinada técnica de modelado, limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.

Esta fase se encuentra relacionada con la fase de modelado, ya que en función de la técnica de modelado elegida, los datos requieren ser procesados de una manera o de otra, por esta razón las fases de preparación y de modelado interactúan de forma permanente.

#### **a. Seleccionar los datos**

En esta etapa se selecciona un subconjunto de los datos adquiridos anteriormente apoyándose en criterios previamente definidos en las fases anteriores como la calidad de los datos en cuanto a su

completitud, corrección de los datos y limitaciones en el volumen o en los tipos de datos que están relacionados con las técnicas de minería de datos seleccionadas.

**b. Limpiar los datos**

Esta tarea complementa a la anterior y es una de las que más tiempo y esfuerzo consume debido a la diversidad de técnicas que pueden aplicarse para optimizar la calidad de los datos a objeto de prepararlos para la fase de modelación. Algunas de las técnicas a utilizar para este propósito son la normalización de los datos, discretización de campos numéricos, tratamiento de valores faltantes y reducción del volumen de datos.

**c. Construir los datos**

Esta tarea incluye las operaciones de preparación de los datos tales como la generación de nuevos atributos a partir de atributos ya existentes, integración de nuevos registros o transformación de valores para atributos existentes.

**d. Integrar los datos**

La integración de los datos implica la creación de nuevas estructuras a partir de los datos seleccionados, por ejemplo, generación de nuevos campos a partir de otros existentes, creación de nuevos registros, fusión de tablas campos o nuevas tablas donde se resumen características de múltiples registros o de otros campos en nuevas tablas de resumen.

**e. Formateo de los datos**

Esta tarea consiste principalmente en la realización de transformaciones sintácticas de los datos sin modificar su significado de tal forma que se permita y se facilite utilizar alguna técnica de minería de datos en concreto, como por ejemplo la reordenación de los campos y/o de los registros de la tabla o el ajuste de los valores de los campos a las limitaciones de las herramientas de modelación (eliminar comas,

tabuladores, caracteres especiales, máximos y mínimos para las cadenas de caracteres, etc.).

#### **4. Modelado**

En esta fase de CRISP-DM se seleccionan las técnicas de modelado más apropiadas para el proyecto de minería de datos específico. Las técnicas a utilizar en esta fase se eligen en función de los siguientes criterios:

- Ser apropiada para el problema.
- Disponer de los datos adecuados.
- Cumplir los requisitos del problema.
- Tiempo adecuado para obtener un modelo.
- Conocimiento de la técnica.

Previamente al modelado de los datos se debe determinar un método de evaluación de los modelos que permita establecer el grado de adecuación de cada uno de ellos. Después de concluir estas tareas genéricas se procede a la generación y evaluación del modelo. Los parámetros utilizados en la generación del modelo dependen de las características de los datos y de las características de precisión que se quieran lograr con el modelo.

##### **a. Escoger la técnica de modelado**

Esta tarea consiste en la selección de la técnica de minería de datos más apropiada al tipo de problema que se quiere resolver. Para esta selección, se debe considerar el objetivo principal del proyecto y la relación con las herramientas de minería de datos existentes. Por ejemplo, si el problema es de clasificación, se podrá elegir de entre árboles de decisión, k-vecinos más cercanos o razonamiento basado en casos (CBR), si el problema es de predicción, análisis de regresión o redes neuronales, o si el problema es de segmentación, redes neuronales, técnicas de visualización, etc.

#### **b. Generar el plan de prueba**

Se debe generar un procedimiento destinado a probar la calidad y validez del modelo elegido una vez que éste esté construido. Por ejemplo, en una tarea supervisada de minería de datos como la clasificación, es común usar la razón de error como medida de la calidad. Entonces, típicamente se separan los datos en dos conjuntos, uno de entrenamiento y otro de prueba, para luego construir el modelo basado en el conjunto de entrenamiento y medir la calidad del modelo generado con el conjunto de prueba.

#### **c. Construir el modelo**

A continuación se ejecuta la técnica seleccionada sobre los datos previamente preparados para generar uno o más modelos. Todas las técnicas de modelado tienen un conjunto de parámetros que determinan las características del modelo a generar. La selección de los mejores parámetros es un proceso iterativo y se basa exclusivamente en los resultados generados. Estos deben ser interpretados y su rendimiento justificado.

#### **d. Evaluar el modelo**

En esta última tarea de esta fase de modelado los ingenieros de minería de datos interpretan los modelos de acuerdo al conocimiento preexistente del dominio y los criterios de éxito preestablecidos. Expertos en el dominio del problema juzgan los modelos dentro del contexto del dominio y expertos en minería de datos aplican sus propios criterios (seguridad del conjunto de prueba, pérdida o ganancia de tablas, etc.).

### **5. Evaluación**

En esta fase se evalúa el modelo, teniendo en cuenta el cumplimiento de los criterios de éxito del problema. Debe considerarse además que la fiabilidad calculada para el modelo se aplica solamente para los datos sobre los que se realizó el análisis. Es preciso revisar el proceso, teniendo en cuenta los



resultados obtenidos, para poder repetir algún paso anterior, en el que se pueda haber cometido algún error. Considerar que se pueden emplear múltiples herramientas para la interpretación de los resultados. Si el modelo generado es válido en función de los criterios de éxito establecidos en la fase anterior, se procede a la explotación del modelo.

#### **a. Evaluar los resultados**

En los pasos de evaluación anteriores se trataron factores tales como la exactitud y generalidad del modelo generado. Esta tarea involucra la evaluación del modelo en relación a los objetivos del negocio y busca determinar si hay alguna razón de negocio para la cual el modelo sea deficiente, o si es aconsejable probar el modelo en un problema real si el tiempo y las restricciones lo permiten. Además de los resultados directamente relacionados con el objetivo del proyecto, ¿es aconsejable evaluar el modelo en relación a otros objetivos distintos a los originales?, esto podría revelar información adicional.

#### **b. Revisar el proceso**

Este proceso se refiere a calificar al proceso entero de minería de datos a objeto de identificar elementos que pudieran ser mejorados.

#### **c. Determinar los próximos pasos**

Si se ha determinado que las fases hasta este momento han generado resultados satisfactorios podría pasarse a la siguiente fase, en caso contrario podría decidirse por hacer otra iteración desde la fase de preparación de los datos o de modelado con distintos parámetros. Podría incluso darse el caso de que en esta fase se decida empezar desde cero con un nuevo proyecto de minería de datos.

### **6. Despliegue o implantación**

En esta fase, y una vez que el modelo ha sido construido y validado, se transforma el conocimiento obtenido en acciones dentro del proceso de negocio, esto puede hacerse por ejemplo cuando el analista recomienda acciones basadas en la observación del modelo y sus resultados, o por ejemplo

aplicando el modelo a diferentes conjuntos de datos o como parte del proceso (en análisis de riesgo de créditos, detección de fraudes, etc.). Generalmente un proyecto de minería de datos no concluye en la implantación del modelo, ya que se deben documentar y presentar los resultados de manera comprensible para el usuario con el objetivo de lograr un incremento del conocimiento. Por otra parte, en la fase de explotación se debe asegurar el mantenimiento de la aplicación y la posible difusión de los resultados.

**a. Planear la implantación**

Para implementar el resultado de la minería de datos en la organización, esta tarea toma los resultados de la evaluación y concluye una estrategia para su implementación. Si un procedimiento general se ha identificado para crear el modelo, este procedimiento debe ser documentado para su posterior implementación.

**b. Planear la monitorización y mantenimiento**

Si los modelos resultantes del proceso de minería de datos son implementados en el dominio del problema como parte de la rutina diaria, es aconsejable preparar estrategias de monitorización y mantenimiento para ser aplicadas sobre los modelos. La retroalimentación generada por la monitorización y mantenimiento pueden indicar si el modelo está siendo utilizado apropiadamente.

**c. Producir el informe final**

Es la conclusión del proyecto de minería de datos realizado. Dependiendo del plan de implementación, este informe puede ser sólo un resumen de los puntos importantes del proyecto y la experiencia adquirida o puede ser una presentación final que incluya y explique los resultados logrados con el proyecto.

**d. Revisar el proyecto**

En esta tarea se evalúa qué cosas se hicieron correctamente y cuáles fueron incorrectas, así como aquellos puntos que se podrían mejorar en el proyecto.

## 1.4 Herramientas y Tecnologías

### ARASAAC

ARASAAC (*AAC Symbols and shared resources - ARASAAC 2022*) ofrece recursos gráficos basado en el uso de pictogramas para facilitar la comunicación y la accesibilidad cognitiva a todas las personas que, por distintos factores (autismo, discapacidad intelectual, desconocimiento del idioma, personas mayores, etc.), presentan graves dificultades en estas áreas, que dificultan su inclusión en cualquier ámbito de la vida cotidiana.

El objetivo fundamental del proyecto es la creación de un sistema pictográfico de comunicación y un conjunto de herramientas de libre distribución, que faciliten la accesibilidad de carácter comunicativo y cognitivo en diversos ámbitos de la vida, a todas las personas que lo puedan requerir (*ARASAAC – CAREI sin fecha*).

ARASAAC está alojado en el Portal Aragonés de Comunicación Aumentativa y Alternativa (*AAC Symbols and shared resources - ARASAAC 2022*), junto con otro tipo de recursos para la comunicación: materiales adaptados, herramientas on line, software, ejemplos de uso... El portal ofrece diversos catálogos gráficos de pictogramas en color, blanco y negro, fotografías, vídeos... En la actualidad estos pictogramas están traducidos a quince idiomas.

Se puede acceder a una serie de herramientas on line que facilitan la creación y elaboración de materiales para la comunicación, como bingos, símbolos, horarios, calendarios, dominós y tableros. Pero existe también un amplio repertorio de materiales elaborados por los propios usuarios que lo convierte en un proyecto colaborativo y abierto (*ARASAAC – CAREI sin fecha*).

### Python 3.7.14

Python (*Python.org sin fecha*) es lenguaje de programación orientado a objetos de propósito general que enfatiza la legibilidad del código a través de su generoso uso de espacios en blanco. Python es fácil de aprender y uno de los favoritos de los programadores y desarrolladores. De hecho, Python es en el momento que se escribe

este trabajo el lenguaje de programación más popular del mundo, seguido de C y Java.

Varias bibliotecas de Python admiten tareas de ciencia de datos, incluidas las siguientes:

- Numpy para manejar arreglos de grandes dimensiones
- Pandas para manipulación y análisis de datos
- Matplotlib para construir visualizaciones de datos

Además, Python es particularmente adecuado para implementar el aprendizaje automático a gran escala. Su conjunto de bibliotecas especializadas de aprendizaje profundo y aprendizaje automático incluye herramientas como scikit-learn, Keras y TensorFlow, que permiten a los científicos de datos desarrollar modelos de datos sofisticados que se conectan directamente a un sistema de producción.

### **Anaconda 2022.05**

Anaconda (Anaconda sin fecha) es una distribución de los lenguajes de programación Python y R para computación científica (ciencia de datos, aplicaciones de aprendizaje automático, procesamiento de datos a gran escala, análisis predictivo, etc.), que tiene como objetivo simplificar la administración y la implementación de paquetes. La distribución incluye paquetes de ciencia de datos adecuados para Windows, Linux y macOS.

La distribución de Anaconda viene con más de 250 paquetes instalados automáticamente, y se pueden instalar más de 7500 paquetes de código abierto adicionales desde PyPI (Indexador de los paquetes de python), así como el paquete conda y el administrador de entorno virtual.

### **Google Colab**

Colaboratory, o “Colab” para simplificar (Google sin fecha), es un producto de Google Research. Colab permite que cualquier persona escriba y ejecute código Python arbitrario a través del navegador, y es especialmente adecuado para el aprendizaje

automático, el análisis de datos y la educación. Más técnicamente, Colab es un servicio de Jupyter Notebook alojado que no requiere configuración para su uso, al tiempo que brinda acceso gratuito a los recursos informáticos, incluidas las GPU (Intel et al. 2022).

### **Spacy 3.4.0**

spaCy (spaCy sin fecha) es una biblioteca para procesamiento avanzado de lenguaje natural en Python y Cython. Se basa en las últimas investigaciones y fue diseñado desde el primer día para ser utilizado en productos reales.

spaCy viene con canalizaciones previamente entrenadas y actualmente admite tokenización y capacitación para más de 60 idiomas. Cuenta con modelos de red neuronal y de velocidad de última generación para etiquetado gramatical, análisis, reconocimiento de entidades nombradas, chunking/fragmentación, clasificación de texto y más, aprendizaje multitarea con transformadores pre entrenados como BERT, así como un sistema de entrenamiento listo para la producción y un modelo sencillo. empaquetado, implementación y gestión del flujo de trabajo. spaCy es un software comercial de código abierto, publicado bajo la licencia MIT.

Algunas características interesantes de spaCy son:

- Soporte para más de 60 idiomas
- Canalizaciones entrenadas para diferentes idiomas y tareas
- Aprendizaje multitarea con transformadores pre entrenados como BERT
- Compatibilidad con vectores de palabras pre entrenadas e embeddings
- Tokenización motivada lingüísticamente
- Componentes para el reconocimiento de entidades nombradas, etiquetado de partes del discurso, análisis de dependencias, segmentación de oraciones, clasificación de texto, lematización, análisis morfológico, vinculación de entidades y más
- Compatibilidad con modelos personalizados en PyTorch, TensorFlow y otros marcos

### **Pandas 1.3.5**

Pandas es una herramienta de código abierto rápida, potente, flexible y fácil de usar, para la manipulación y análisis de datos construida sobre el lenguaje de programación Python (*pandas documentation 2022*).

### **HNSWLIB**

La búsqueda aproximada del vecino más cercano es un paso importante en la generación eficiente de vectores de documentos similares para un vector de consulta dado (Agarwal, Koniashev, Schaefer 2021). Hnswlib (Malkov, Yashunin 2018) es un paquete de búsqueda rápida del vecino más cercano aproximado.

### **SentenceTransformers**

Según (Reimers, Gurevych 2019) SentenceTransformers es un marco de trabajo de Python de última generación para embeddings (incrustaciones) de oraciones, texto e imágenes. Se puede usar este marco de trabajo para calcular embeddings de oraciones/textos para más de 100 idiomas. Estas incrustaciones se pueden comparar, por ejemplo, con similitud de coseno para encontrar oraciones con un significado similar. Esto puede ser útil para la búsqueda semántica de texto similar.

El marco se basa en PyTorch y Transformers y ofrece una gran colección de modelos previamente entrenados ajustados para diversas tareas. Además, es fácil ajustar sus propios modelos (*SentenceTransformers Documentation 2022*).

### **Lucidchart**

El mapeo del flujo de trabajo puede generar resultados valiosos dondequiera que se utilicen procesos complejos (Faulkner, Contributor 2018).

Lucidchart es una aplicación en línea diseñada para la creación rápida y fácil de diagramas de flujo, utilizando una interfaz intuitiva de arrastrar y soltar y una gran cantidad de elementos de diseño de gráficos integrados. Debido a que es un software basado en navegador, no se requiere descarga y Lucidchart se puede usar desde cualquier tipo de computadora (Faulkner, Contributor 2018).

Según (Rueda, Estupiñán 2018) el 50.00% de los universitarios, la aplicación Lucidchart es totalmente útil en la asignatura Desarrollo de aplicaciones. Asimismo,

el 54.55% de los encuestados consideran que este servicio en la nube es totalmente innovador para la Licenciatura en Gestión de Negocios y Tecnologías de la Información. Cabe mencionar que la interfaz de la aplicación Lucidchart es sencilla, rápida, agradable y comprensible.

### **1.5 Conclusiones Parciales**

Con el estudio de los referentes teóricos sobre los sistemas de comunicación aumentativa y alternativa realizado en este capítulo se evidenció la importancia que tienen en el desarrollo de las personas con dificultad del habla, y del uso de un analizador semántico en estos sistemas. El análisis de los sistemas homólogos permitió detectar las funcionalidades básicas de un pictotraductor y del analizador semántico. Además, demostró la necesidad de desarrollar un analizador semántico para los Sistemas de CAA Pictotraductor para facilitar a las personas con dificultades del habla en Cuba y a sus familiares una herramienta que les ayude a promover las competencias y habilidades de estas personas. Por último, el estudio de herramientas y tecnologías permitió seleccionar las más adecuadas para la implementación del analizador semántico.

## **CAPÍTULO 2: ANALIZADOR SEMÁNTICO: DISEÑO, CONSTRUCCIÓN Y METODOLOGÍA**

En este capítulo se presenta detalladamente la propuesta de solución para mejorar las traducciones de los Sistemas de Comunicación Aumentativa y Alternativa basado en las técnicas de Procesamiento del Lenguaje Natural. Este trabajo implementa una solución basada en un motor de búsqueda semántica usando los embeddings de SBERT una adaptación del modelo BERT(Devlin et al. 2018) propuesto por (Reimers, Gurevych 2019). La construcción de la solución propuesta fue guiada por las fases de la metodología CRISP-DM. En el presente capítulo se mostrarán los artefactos generados desde la fase uno hasta la fase cinco.

### **2.1 Comprensión del negocio/problema**

En esta primera fase del proceso de desarrollo de la solución se busca comprender el problema, buscar las soluciones existentes al problema, determinar los objetivos y hacer el plan del proyecto (*IBM SPSS Modeler CRISP-DM Guide 2022*).

Como se ha mencionado anteriormente, el objetivo del presente trabajo es construir un analizador semántico para un pictotraductor, sin embargo, existen varias situaciones que son necesarias abordar para lograr una mejor comprensión del problema.

Las palabras homógrafas se escriben igual, pero tienen significados diferentes, por ejemplo: la palabra “banco” tiene disímiles acepciones en el diccionario de la lengua española. A continuación, se muestran cinco de ellas:

1. m. Asiento, con respaldo o sin él, en el que pueden sentarse dos o más personas.
2. m. Madero grueso escuadrado que se coloca horizontalmente sobre cuatro pies y sirve de mesa para labores de carpinteros y otros artesanos.
3. m. En los mares, ríos y lagos navegables, bajo que se prolonga en una gran extensión.
4. m. Conjunto de peces que van juntos en gran número.



5. m. Empresa dedicada a realizar operaciones financieras con el dinero procedente de sus accionistas y de los depósitos de sus clientes.

Es por ello, que la palabra “banco” se puede encontrar en varios contextos y, por tanto, al traducir dicha palabra a pictograma, esta debe hacer alusión a imágenes muy diferentes.

Por otro lado, las familias de palabras son un conjunto de palabras que tienen en común una porción de la palabra. A esta parte comunitaria se le llama lexema o raíz, mientras que al resto de la palabra se le llama morfema. En ocasiones es importante extraer el lexema de la palabra para poder encontrar un pictograma adecuado para representar dicha palabra.

Otro elemento importante lo constituyen los nombres propios. Los nombres propios son palabras que se utilizan para mencionar lugares, eventos, negocios, animales y personas puntuales. Se trata siempre de entidades únicas. Por ejemplo: La Palma en Arroyo Naranjo se refiere a un lugar, no a la palma como planta, ni a la palma de la mano. En consecuencia, la representación de esta palabra en pictograma puede hacer alusión a imágenes diferentes.

En consecuencia, con los elementos analizados se proponen como tareas del procesamiento del lenguaje natural los siguientes pasos:

1. Tokenización
2. Etiquetado gramatical
3. Lematización
4. Fragmentación / Reconocimiento de entidades nombradas
5. Construcción de modelo con la base de datos de ARASAAC.
6. Selección de pictogramas según el texto de entrada.

Antes de comenzar este análisis lingüístico es necesario comprender cómo está organizada la base de datos ARASAAC.

## 2.2 Análisis exploratorio de la base de datos ARASAAC

Según (*IBM SPSS Modeler CRISP-DM Guide 2022*) la comprensión de los datos implica acceder a los datos y explorarlos mediante tablas y gráficos . Esto permite determinar la calidad de los datos.

Primero se procede a la recopilación de los datos necesarios para llevar a cabo el estudio de este trabajo desde la API de ARASAAC. Estos datos son referentes a pictogramas que incluyen variadas informaciones como puede ser, los identificadores, nombres, descripciones, categorías, etiquetas, etc., en total son 19 características por pictograma (Figura 4), es importante señalar que sólo las palabras claves se muestran en español. Estos datos fueron almacenados como un dataframe de pandas en un fichero "csv", donde las columnas representan las características y las filas los registros/pictogramas (Figura 5).

Después de realizar el estudio de los datos, es relevante destacar que algunas descripciones y nombres son las mismas para diferentes pictogramas. Por ejemplo: los pictogramas en la Figura 6 tienen el mismo nombre "cocina", como se puede observar, de los cinco pictogramas solo el tercero y el quinto se parecen, los restantes son diferentes, es decir, se tienen 4 pictogramas diferentes con el mismo nombre. Otro aspecto a analizar en estos pictogramas es cuán diferentes son, tres de ellos representan un fogón, uno representa un sartén y uno representa una cocina (como lugar o habitación de una casa). Para la solución propuesta esto representa un problema, puesto que puede generar traducciones donde el contexto estará mal interpretado.

Cuando se observan las descripciones de los pictogramas de la Figura 6, se encuentra un problema similar al que tienen los nombres de los pictogramas. Todos tienen la misma descripción "*Pieza de la casa en donde se prepara la comida.*", con excepción del último que su descripción es "*Aparato para guisar o calentar los alimentos, que funciona con gas, carbón, electricidad, etc.: he limpiado los fuegos de la cocina con amoníaco.*". Cabe señalar que el tercer pictograma tiene una imagen similar a la del quinto pictograma, por tanto, además de descripciones iguales para pictogramas de imágenes diferentes, también existen descripciones diferentes para pictogramas de imágenes similares.

```

{
  "_id": 10752,
  "created": "2009-11-21T17:23:36.000Z",
  "downloads": 0,
  "tags": [
    "place",
    "building",
    "home",
    "room",
    "communication",
    "signaling system",
    "core vocabulary"
  ],
  "synsets": [
    "03625099-n"
  ],
  "sex": false,
  "lastUpdated": "2021-02-06T06:44:02.591Z",
  "schematic": true,
  "keywords": [
    {
      "keyword": "cocina",
      "type": 2,
      "meaning": " f. Pieza de la casa en donde se prepara la comida.",
      "plural": "cocinas",
      "hasLocution": true
    }
  ],
  "categories": [
    "room",
    "signaling system",
    "core vocabulary-place",
    "core vocabulary-communication"
  ],
  "violence": false,
  "hair": false,
  "skin": false,
  "aac": true,
  "aacColor": false
},

```

Figura 4. Características de los pictogramas de la API de ARASAAC (AAC Symbols and shared resources - ARASAAC 2022)

index	id	keyword	meaning	categories	tags
0	0	2247	acera f. acera, aceras: Orilla de la calle o de otra...	['infrastructure', 'land transport', 'road saf...]	['place', 'infrastructure', 'movement', 'traff...
1	1	2247	vereda vereda, veredas.	['infrastructure', 'land transport', 'road saf...]	['place', 'infrastructure', 'movement', 'traff...
2	2	2248	agua f. agua, aguas: Sustancia cuyas moléculas está...	['beverage', 'mineral rich food']	['feeding', 'food', 'beverage', 'mineral rich ...]
3	3	2249	alfombra f. alfombra, alfombras: Tejido de lana o de ot...	['trousseau']	['object', 'household', 'trousseau', 'home']
4	4	2250	almohada f. almohada, almohadas: Colchoncillo para recl...	['trousseau', 'core vocabulary-object']	['object', 'household', 'trousseau', 'home', '...]

Figura 5. Ejemplo de dataframe de pandas



Figura 6. Ejemplo de pictogramas con el mismo nombre (cocina).

La solución de este proyecto requiere que los pictogramas estén bien descritos, sus nombres y descripciones solo deben ser iguales cuando los pictogramas representan imágenes similares, por ejemplo, no hay ninguna contradicción en que la palabra “fogón” sea usada para los pictogramas 2, 3 y 5 de la Figura 6, puesto que los tres representan lo mismo. Sin embargo, los nombres y las descripciones deben expresar lo que realmente representa el pictograma, un ejemplo donde no se corresponde el nombre con la imagen se puede observar en el primer pictograma de la Figura 6 donde aparece un sartén pero el nombre que tiene asociado este picto es cocina.

Después de hacer la exploración inicial de los datos se puede afirmar que estos cubren parcialmente las necesidades básicas para cumplir los objetivos del proyecto. Por tanto se seleccionan únicamente las características necesarias para la modelación, de las 19 características por pictogramas (Figura 4) de ARASAAC, se utilizan únicamente las siguientes características: id, nombre y descripción; debido a que el resto de las características están en inglés. Una vez seleccionadas dichas características se procede a eliminar todos los registros donde el id o el nombre o la descripción es nulo, obteniendo un registro sin valores nulos. Terminada esta fase crucial se procede a explicar el análisis lingüístico de la propuesta de solución.

### 2.3 Análisis Lingüístico

El análisis lingüístico es crucial para estudiar y entender la estructura, la forma y las características lingüísticas del texto (Kahramonovna 2021). En este subepígrafe se realiza el análisis lingüístico aplicado a la solución para mejorar los resultados del mecanismo de búsqueda.

## Tokenización

La tokenización es el primer paso en la mayoría de los trabajos de procesamiento de texto (Park et al. 2020). La tokenización se utiliza para convertir textos en unidades más pequeñas (tokens) (Khekare, Verma 2020). Un token equivale a una palabra, de modo que la oración “Mamá está en la cocina” está compuesta por 5 tokens. Estos tokens son posteriormente pasado para el etiquetado gramatical.

## Etiquetado Gramatical

En el PLN, el etiquetado de parte del discurso (Part of Speech - PoS), o también etiquetado gramatical, es una tarea de clasificación de tokens que predice la etiqueta gramatical de cada palabra en contexto (Lim, Park 2020). Además, es el proceso en el que a cada palabra de una oración se le asigna una categoría gramatical (Deshmukh, Kiwelekar 2020). En la Figura 7, se observa un ejemplo para la oración “Mamá está en la cocina” donde cada palabra recibe su clase gramatical dependiendo del contexto de la oración, donde, PROPN significa Nombre propio, VERB significa verbo, ADP significa preposición, DET significa artículo definido, NOUN significa sustantivo (*Universal POS tags 2022*).

Mamá	está	en	la	cocina
PROPN	VERB	ADP	DET	NOUN

Figura 7. Ejemplo de etiquetado gramatical.

El etiquetado gramatical es importante en la solución de esta investigación en dos etapas, para identificar los Chunks/fragmentos que se explican más adelante y al final para determinar cuál de los dos embeddings (nombres o descripciones) se debe usar como mecanismo de búsqueda dependiendo de la categoría gramatical que tenga la palabra. Se usará los embeddings de los nombres para los artículos definidos, preposiciones, verbos, sustantivos y pronombres y para las demás categorías gramaticales se usará los embeddings de las descripciones.

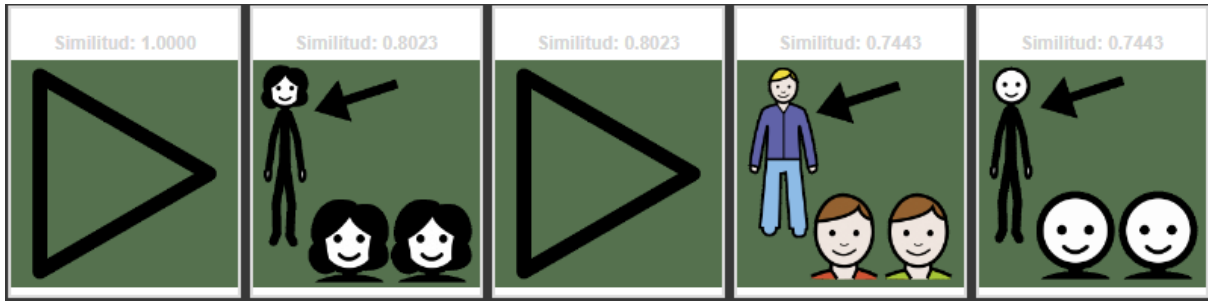


Figura 8. Los 5 primeros resultados de la búsqueda para la cadena “el” usando los embeddings de los nombres.

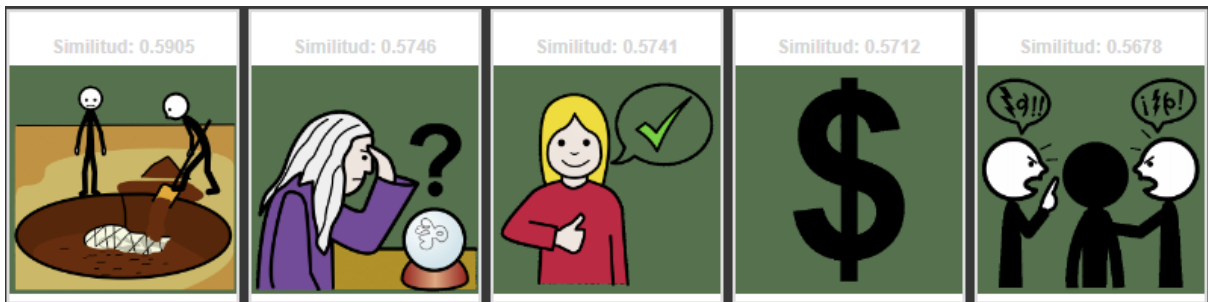


Figura 9. Los 5 primeros resultados de la búsqueda para la cadena “el” usando los embeddings de las descripciones.

En la figura 8 se observa como el embedding de los nombres presenta mejor resultado en relación al embedding de las descripciones (figura 9). El primer pictograma de la figura 8, que es el mismo que el tercero, es el ideal para el artículo definido “el”, los demás tienen una relación léxica semántica con el mismo o equivalen al pronombre personal “él”. Sin embargo, en la figura 9 no se tiene ningún pictograma que se pueda usar para la palabra en cuestión. Por este motivo se asume que el embedding de los nombres es el adecuado para las búsquedas de palabras con la categoría gramatical “artículo definido”.

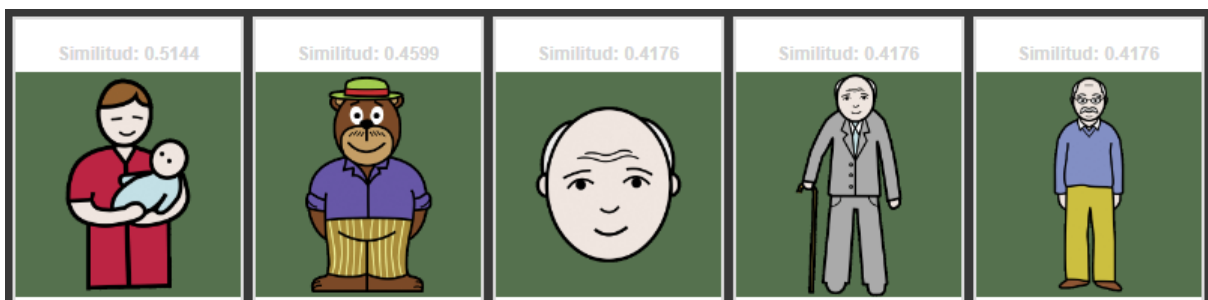


Figura 10. Los 5 primeros resultados de la búsqueda para la cadena “papá” usando los embeddings de las descripciones.

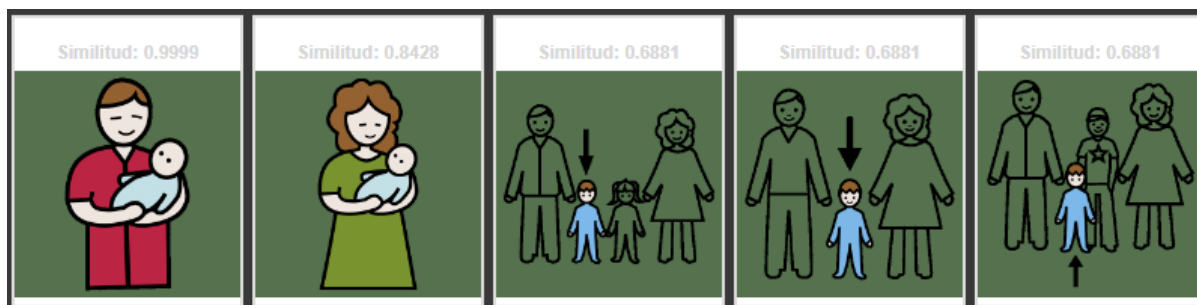


Figura 11. Los 5 primeros resultados de la búsqueda para la cadena “papá” usando los embeddings de los nombres.

En la figura 10 se observa como el embedding de las descripciones presenta mejor resultado en relación al embedding de los nombres (figura 11) para ilustrar sustantivos propios. Aunque en la primera posición de la figura 11 se tiene el pictograma ideal para la palabra “papá”, la figura 10 contiene el pictograma ideal y además mayor cantidad de entidades paternas. Por este motivo se asume que los embeddings de las descripciones son los adecuados para las búsquedas de palabras con la categoría gramatical “sustantivos propios”.

### Lematización

La lematización (Khyani et al. 2021) es el proceso de encontrar la forma morfológica básica de una palabra. Un lematizador utiliza una base de conocimiento de sinónimos de palabras y terminaciones de palabras para garantizar que solo las palabras que significan cosas similares se consoliden en un solo token; este token es el utilizado para el análisis. Por ejemplo, en inglés “am,” “are,” “is,” “was,” “were”, se tratarán todos de la misma manera que "be" por un lematizador, aunque tengan significados diferentes. En la tabla 3 se observan algunas conjugaciones verbales de verbos en español y sus respectivos lemas o raíces.

Tabla 3. Formas flexionadas y sus lemas/raíz

Palabras gráficas	Lema
somos, soy, eres, fueron	ser
voy, fui, irán	ir
poder, pudieron, podrán	poder
estaba, estuve, estaban	estar

Los nombres de los pictogramas en ARASAAC no usan conjugaciones de verbos, eso implica que los los embeddings de los nombres, el ideal para la búsqueda de palabras con categoría verbal, no presenta buenos resultados cuando la cadena de búsqueda es una palabra que sea la forma verbal de un verbo, es decir, conjugaciones. Por ejemplo: en la Figura 12 se tiene una conjugación del verbo preferir “prefiero” y a la derecha su lema la palabra “preferir”, si se coloca la palabra “prefiero” en el buscador sin antes hacer la lematización no se obtienen buenos resultados, como se observa en la figura 13. Sin embargo, cuando se realiza la lematización antes de buscar la palabra se obtienen mejores resultados, como se observa en las figura 14.

Token	Lemma	POS	Morphological Features
Yo	yo	PRON	Case=Nom Number=Sing Person=1 PronType=Prs
prefiero	preferir	VERB	Mood=Ind Number=Sing Person=1 Tense=Pres VerbForm=Fin
el	el	DET	Definite=Def Gender=Masc Number=Sing PronType=Art
vuelo	vuelo	NOUN	Gender=Masc Number=Sing
de	de	ADP	
la	el	DET	Definite=Def Gender=Fem Number=Sing PronType=Art
mañana	mañana	NOUN	Gender=Fem Number=Sing
para	para	ADP	
la	el	DET	Definite=Def Gender=Fem Number=Sing PronType=Art
Habana	Habana	PROPN	

Figura 12. Etiquetado gramatical de la oración “Yo prefiero el vuelo de la mañana para la habana”.

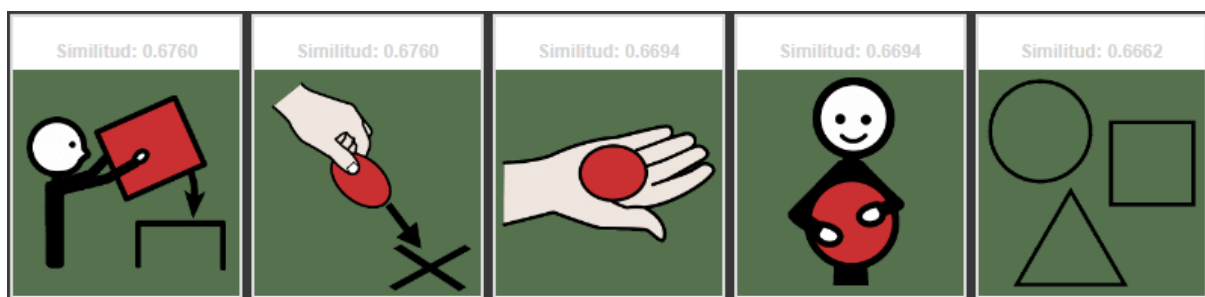


Figura 13. Los 5 primeros resultados de la búsqueda para la palabra “prefiero” usando los embeddings de los nombres.

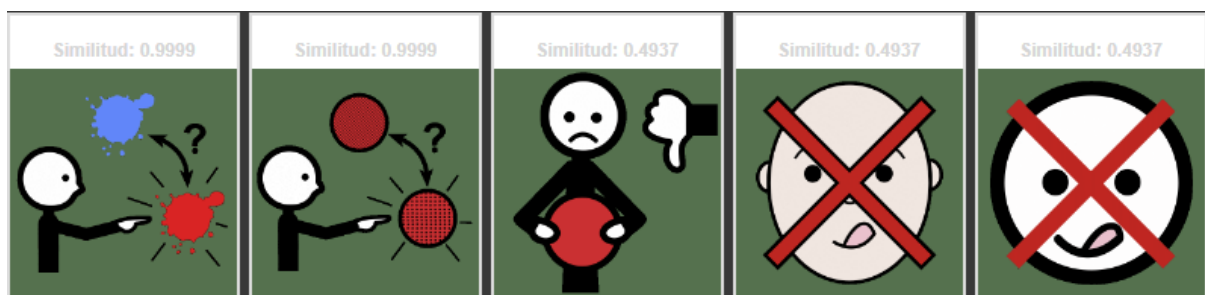


Figura 14. Los 5 primeros resultados de la búsqueda para la palabra “preferir” usando los embeddings de los nombres.



## Fragmentación (Chunking)

La fragmentación de texto se refiere a la tarea de dividir una cadena de información textual en grupos de unidades relacionadas que no se superponen. Este es un problema fundamental en numerosos campos donde es relevante comprender la relación entre unidades de datos secuenciales (Borrelli, Gongora Svartzman, Lipizzi 2020). En lugar de simples símbolos que pueden no representar el significado real del texto, es recomendable utilizar frases como "La Habana" como una sola palabra en lugar de las palabras separadas "La" y "Habana".

La fragmentación utiliza el etiquetado gramatical como entrada y proporciona fragmentos como salida. La fragmentación es muy importante cuando se desea extraer información de texto como ubicaciones, nombres de personas, etc. En PLN a este proceso se le llama extracción/reconocimiento de entidades nombradas (Yadav, Bethard 2019).

La fragmentación es una etapa opcional en la propuesta de solución, dependerá si el usuario pretende representar frases telegráficas o no. Por ejemplo, la oración "*Yo prefiero el vuelo de la mañana*" al ser traducida sin pasar por la fragmentación se obtienen 7 pictogramas como salida, un pictograma para cada token/palabra (figura 15); sin embargo, si se pasa antes por la etapa de fragmentación se obtienen 5 pictogramas como salida, puesto que "el vuelo" y "la mañana" al pasar por la fragmentación serían reconocidos como un token (figura 16).

Cabe recalcar que para cada búsqueda de un token se pudiera retornar más de un pictograma, dejando al criterio del usuario esta opción: cuántos pictogramas desea recibir por cada token.

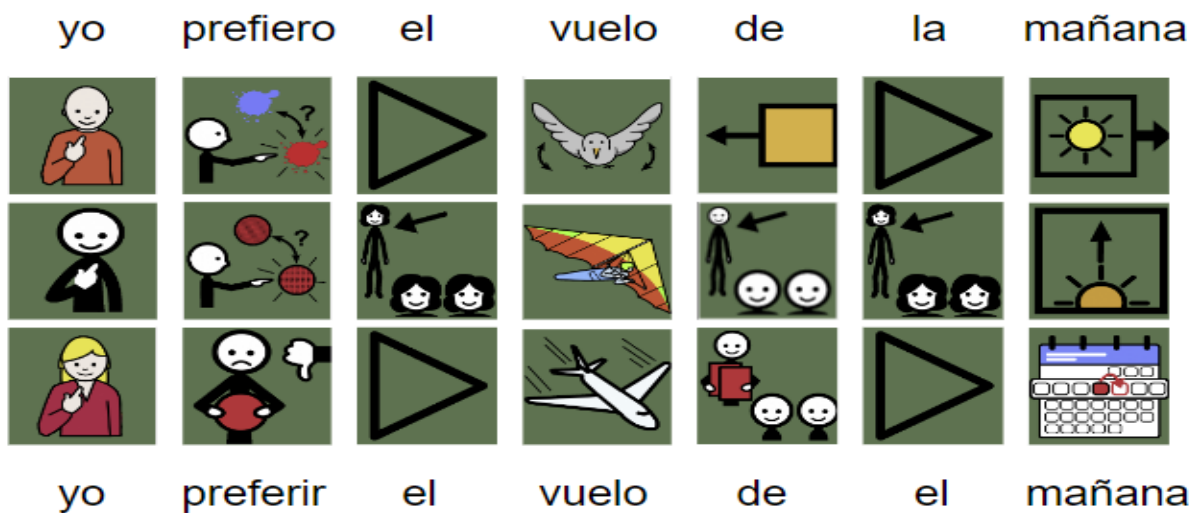


Figura 15. Traducción de la oración “Yo *prefiero* el vuelo de la mañana” sin hacer fragmentación.



Figura 16. Traducción de la oración “Yo *prefiero* el vuelo de la mañana” teniendo en cuenta la fragmentación.

Como se puede observar en las figuras 20 y 21, la búsqueda en este caso retorna 3 pictogramas por cada cadena de búsqueda. Los pictogramas están ordenados descendientemente por la similitud semántica, siendo los de la primera fila los de mayor similitud semántica. Es importante recordar que la similitud depende fuertemente de la descripción o del nombre del pictograma, una pobre o mala descripción afecta considerablemente la traducción, como se muestra en el apartado del análisis exploratorio de los datos y comprensión de los datos.

A continuación, se procede a explicar el entrenamiento y la generación del modelo y por último se evalúa si dicho modelo ha cumplido los criterios de éxito o no.

## **2.4 Modelado**

Debido a que se va implementar un mecanismo de búsqueda semántica, se debe utilizar una técnica con un enfoque basado en la similitud semántica. De los diferentes algoritmos, modelos y técnicas existentes para la similitud semántica, se ha escogido SBERT(Reimers, Gurevych 2019) una adaptación del modelo BERT(Devlin et al. 2018) pre-entrenado.

Seguidamente se explica en qué consiste la búsqueda semántica, además se describe la estructura del modelo BERT y se muestra cómo se utiliza en esta investigación para construir el mecanismo de búsqueda semántica de pictogramas y darle solución al problema planteado, posteriormente se evidencia cómo se integró el análisis lingüístico en el mecanismo de búsqueda.

### **2.4.1 Búsqueda Semántica**

Las medidas de similitud de texto tienen un rol muy importante en la investigación relacionada con el texto. Hay varias aplicaciones o áreas donde se usan las medidas de similitud de texto. Estas áreas son recuperación de información, detección de plagio, búsqueda, agrupación, categorización de texto, detección de temas, sesión de preguntas y respuestas, traducción automática, resumen de texto, etc (Pradhan et al. sin fecha). Como se puede ver las medidas de similitud de texto tienen varias aplicaciones, la búsqueda es una de las más importantes. Según (Pradhan et al. sin fecha) la búsqueda se puede hacer basándose en la similitud. En el presente contexto, la similitud es un proceso donde se determina la relación entre fragmentos de textos. La similitud de texto se define de dos maneras: la similitud léxica y la similitud semántica. La similitud léxica proporciona la similitud sobre la base de la coincidencia de caracteres o declaraciones, por ejemplo "Cuba" y "Cuda" son léxicos similares entre sí. Mientras que la similitud semántica proporciona la similitud sobre la base del significado, por ej. "Universidad de las Ciencias Informáticas" y "UCI" son semánticamente similares entre sí. La palabra "Semántica" (Roy et al. 2019) se refiere al significado o resumen de algo. La "semántica", aplicada a la búsqueda, se refiere esencialmente al estudio de las palabras y su lógica y significado. Es una técnica de

búsqueda de datos, pero no es solo una consulta de búsqueda para encontrar palabras clave, sino también para determinar la intención y el significado contextual de las palabras de búsqueda que utiliza una persona. Por lo tanto, proporciona resultados de búsqueda más significativos al evaluar y comprender la frase de búsqueda, y descubrir los resultados más apropiados en un sitio web, una base de datos o cualquier otro almacén de datos.

Según (Uppal, Rana, Sharma 2019) el motor de búsqueda semántica tiene la capacidad de comprender el significado de la consulta del usuario y, sobre la base de este significado, puede recuperar el resultado o el documento exacto de la web o de cualquier base de datos. Los motores de búsqueda semánticos se basan en el contexto y el concepto de la frase buscada. Proporcionan un resultado más significativo al comprender y evaluar la consulta de búsqueda y encontrar los datos más relevantes en la web.

#### **2.4.2 BERT**

Se ha escogido BERT para dar solución al problema planteado en esta investigación, para entender su funcionamiento es necesario antes presentar en qué consisten los embeddings de palabras y los transformers.

#### **Vectorización (Word Embeddings)**

Las computadoras no pueden procesar los textos en su formato original. Es sabido que las computadoras funcionan con representaciones numéricas, si se profundiza más funcionan con señales eléctricas. Para procesar textos es necesario convertirlos en números.

Existen diferentes métodos de representar una palabra en un conjunto de texto de modo que la computadora lo pueda procesar, a menudo se representa con una codificación one-hot (one-hot encoding), que asigna cada palabra en un vocabulario, que consta de  $V$  elementos únicos, a un índice único en un vector. Por lo tanto, una palabra está representada por un vector donde cada posición está representada por un cero excepto por un uno en la posición apropiada. Los vectores one-hot para la oración “Mamá está en la cocina” son los siguientes:

Mamá = [1, 0, 0, 0, 0]

está = [0, 1, 0, 0, 0]

en = [0, 0, 1, 0, 0]

la = [0, 0, 0, 1, 0]

cocina = [0, 0, 0, 0, 1]

Sin embargo, existen múltiples limitaciones de este tipo de representaciones minimalistas, ya que no captan las características de la palabra y pueden ser bastante grandes según el tamaño del corpus. Con el objetivo de solucionar esta problemática surge la idea de los word embeddings.

Históricamente, la ingeniería de atributos (feature engineering) en el procesamiento del lenguaje natural (PLN) implicaba la creación de funciones numéricas específicas para representar aspectos destacados del texto, como la proporción de sustantivos a pronombres. Este enfoque a menudo requería un conocimiento de dominio significativo y un esfuerzo para identificar características significativas. Por el contrario, los word embeddings se pueden aprender directamente de un corpus de texto y no requieren ningún etiquetado manual o feature engineering, es decir, se pueden aprender sin supervisión (Khattak et al. 2019).

(Brownlee 2019) explica que Word Embeddings consiste en un conjunto de lenguajes de modelado y técnicas de aprendizaje donde las palabras que tienen un mismo significado semántico tienen una representación similar. En la Figura 17 se observa que los vectores están agrupados según una característica en común.

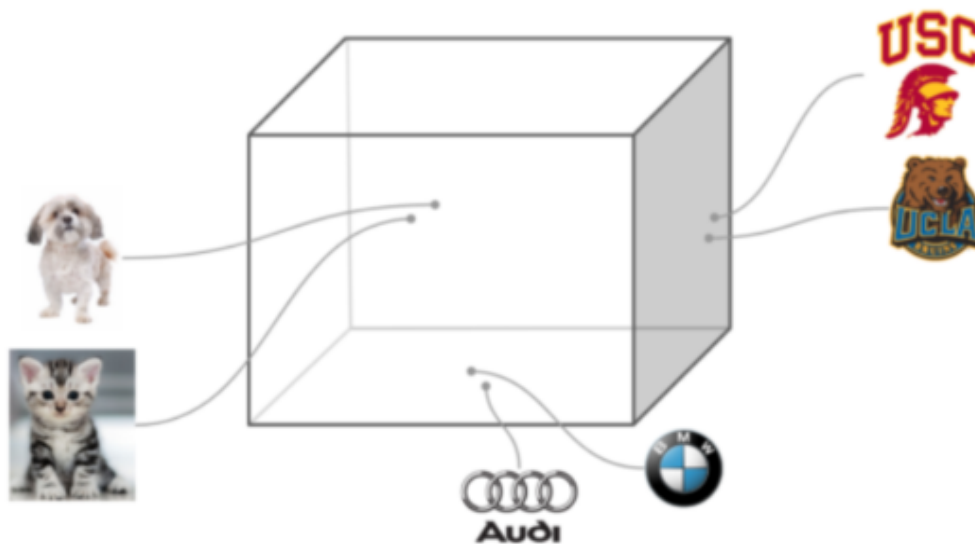


Figura 17. Representación de Palabras en un Espacio Vectorial (Zorrilla Villanueva 2018).

La 'hipótesis distributiva' establece que las palabras que ocurren en los mismos contextos tienen significados similares o relacionados (Khattak et al. 2019). Por lo tanto, se espera que los embeddings de palabras relacionadas semántica o sintácticamente estén más cerca entre sí que las palabras no relacionadas en el espacio vectorial, ver Figura 18. Esta relación depende completamente de los datos del texto, o corpus, de los que se derivan estos embeddings.

Un ejemplo muy famoso de cómo las incrustaciones de palabras pueden representar dicha relación es que se puede hacer un cálculo vectorial como este:

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

Sí, "queen" es como "king", ¡excepto que es una mujer, en lugar de un hombre!

Eso quiere decir que la palabra "king" y "queen" son semánticamente similares, lo mismo acontece con "man" y "woman".

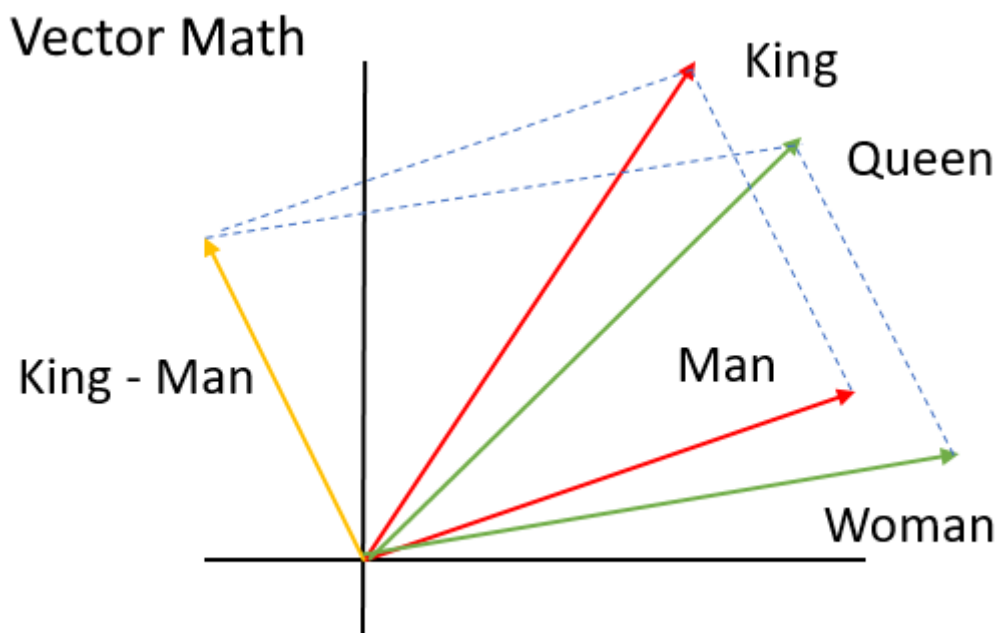


Figura 18. Analogía entre palabras (Zorrilla Villanueva 2018)

En la figura 18 se exhibe una propiedad de Word Embeddings: las analogías entre palabras, donde las relaciones lineales entre pares de palabras son indicadores de la similitud que existe entre estas palabras, de tal manera que es posible establecer una métrica de comparación.

Como se mencionó anteriormente, los word embeddings son representaciones vectoriales de longitud fija para palabras. Existen múltiples formas/métodos de obtener dichas representaciones, dependiendo de la estrategia que se use para la representación, estos métodos son categorizados en dos grupos: modelos basados en predicciones y modelos basados en conteo (Almeida, Xexéo 2019). Muchos de los avances sugeridos (métodos y algoritmos) que se ven en la literatura se han incorporado en kits de herramientas ampliamente utilizados, los más conocidos y avanzados son Word2Vec, gensim, FastText y GloVe, dando como resultado incrustaciones de palabras cada vez más precisas y rápidas, listas para usarse en tareas de Procesamiento de Lenguaje Natural (Almeida, Xexéo 2019).

### **Transformador (Transformer)**

Muchos de los métodos anteriores para el PLN, como word2vec [(Mikolov et al. 2013), (Mikolov et al. 2017), (Goldberg, Levy 2014)] y GloVe (Pennington, Socher, Manning 2014), solían aprender una representación para cada palabra. Sin embargo, para

comprender cómo se relacionan las palabras entre sí, se puede tener un embedding compuesto donde las composiciones de palabras también tienen alguna representación de embedding (Cheng, Dong, Lapata 2016). Por ejemplo, la figura 19 muestra una oración que destaca la relación de las palabras, el color rojo representa la palabra actual que se está fijando, el azul representa los recuerdos y el sombreado indica el grado de activación de la memoria. En otras palabras, la figura 19 muestra que al leer una palabra en una oración, qué palabras anteriores en la oración se recuerdan más (en este caso por los modelos como word2vec y GloVe). Esta relación de palabras muestra que se necesita tener un embedding compuesto para embedding de lenguaje natural. En palabras simples, el gran problema que tienen los métodos como word2vec y GloVe es que se olvidan de las relaciones entre las palabras mientras van leyendo la oración.

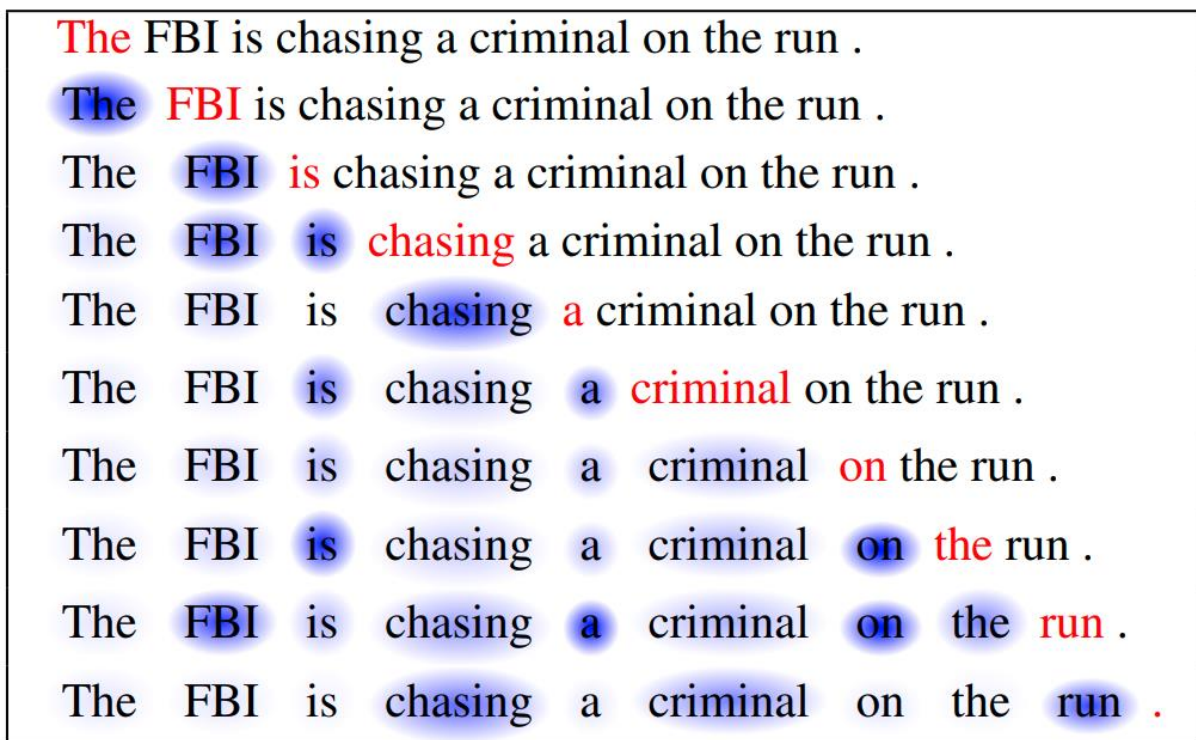


Figura 19. La relación de palabras en una oración que plantea la necesidad de incrustaciones compuestas. (Cheng, Dong, Lapata 2016).

Con el objetivo de resolver los problemas anteriores, el mecanismo de atención se introdujo por primera vez en la traducción automática neuronal por (Bahdanau, Cho, Bengio 2014) para abordar el problema de cuello de botella que surge con el uso de un vector de codificación de longitud fija, donde el decodificador tendría acceso limitado a la información proporcionada por la entrada. Se cree que esto se vuelve



especialmente problemático para secuencias largas y/o complejas, donde la dimensionalidad de su representación se vería obligada a ser la misma que para secuencias más cortas o más simples.

Transformer (Vaswani et al. 2017) es una arquitectura modelo que evita la recurrencia y, en cambio, se basa completamente en el mecanismo de atención para dibujar dependencias globales entre la entrada y la salida. Es un modelo muy destacado de aprendizaje profundo que ha sido ampliamente adoptado en varios campos, como el procesamiento del lenguaje natural (PLN), la visión por computadora (CV en sus siglas en inglés) y el procesamiento del habla. Transformer se propuso originalmente como un modelo de secuencia a secuencia (Sutskever, Vinyals, Le 2014) para la traducción automática. Trabajos posteriores muestran que los modelos pre entrenados (PTMs en sus siglas en inglés) basados en transformadores (Qiu et al. 2020) pueden lograr actuaciones de vanguardia en varias tareas. Como consecuencia, Transformer se ha convertido en la arquitectura de referencia en el Procesamiento del Lenguaje Natural.

Habiendo introducido los word embeddings y el transformer, se procede a explicar el modelo BERT para dar solución al problema de esta investigación.

### **¿Qué es BERT?**

Representaciones de Codificador Bidireccional de Transformadores o BERT (Devlin et al. 2018) es un modelo de representación de palabras contextualizado basado en un codificador de transformador bidireccional multicapa, donde la red neuronal del transformador utiliza capas de atención paralelas en lugar de recurrencia secuencial (Khattak et al. 2019). En otras palabras, se construye utilizando bloques codificadores de transformadores.

BERT sigue siendo uno de los mejores modelos para diferentes tareas de PLN, como la “respuesta a preguntas” (Qu et al. 2019), la “comprensión del lenguaje natural” (Dong et al. 2019), el “análisis de sentimientos” y la “inferencia del lenguaje” (Song et al. 2020).

Según (Kobayashi et al. 2020) BERT utiliza la técnica de modelado de lenguaje enmascarado. Enmascara el 15% de las palabras en el documento/corpus de entrada

y le pide al modelo que prediga las palabras que faltan. Como muestra la Figura 20, a cada bloque de codificación del transformador de la pila se le asigna una oración con una palabra faltante y se supone que el bloque predice la palabra faltante. Es una forma no supervisada porque cualquier palabra puede enmascarse en una oración y se supone que la salida es esa palabra. Como no está supervisado y no requiere etiquetas, los enormes datos de texto de Internet se pueden usar para entrenar el modelo BERT, donde las palabras se seleccionan al azar para enmascararlas.

Se debe tener en cuenta que BERT aprende a predecir la palabra que falta en función de la atención a sus palabras anteriores y siguientes, es bidireccional. Por lo tanto, BERT condiciona conjuntamente tanto el contexto izquierdo (anterior) como el derecho (siguiente) de cada palabra. Además, como la palabra que falta se predice en función de las otras palabras de la oración, las incrustaciones BERT para palabras son incrustaciones conscientes del contexto. Por lo tanto, a diferencia de word2vec [(Mikolov et al. 2013), (Mikolov et al. 2017), (Goldberg, Levy 2014)] y GloVe (Pennington, Socher, Manning 2014) que proporcionan una incrustación única para cada palabra, cada palabra tiene diferentes incrustaciones de BERT en varias oraciones. Los embeddings de BERT para las palabras difieren en diferentes oraciones según su contexto. Por ejemplo, la palabra "banco" tiene diferentes significados y, por lo tanto, diferentes embeddings en las oraciones "El dinero está en el banco" y "Bento está sentado en el banco".

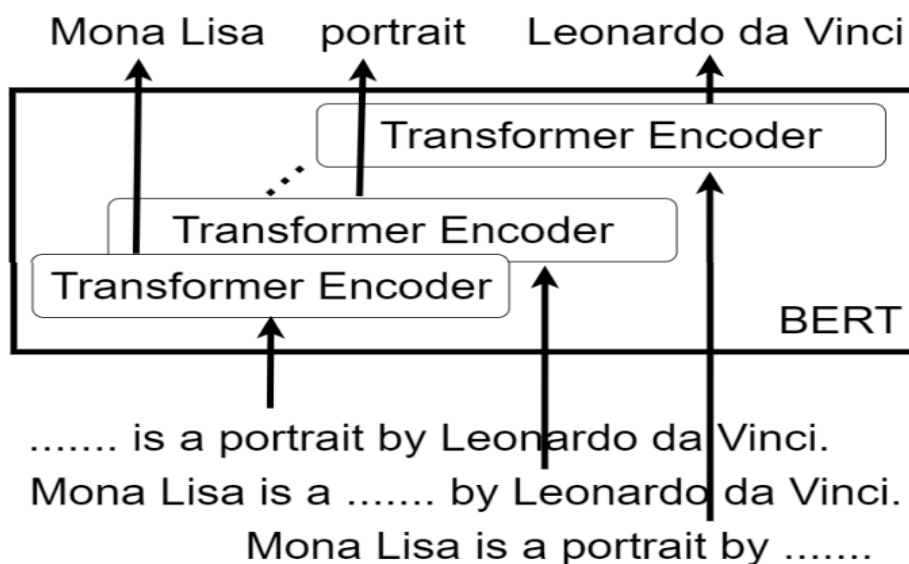


Figura 20. Alimentando oraciones con palabras faltantes al modelo BERT para el entrenamiento (Kobayashi et al. 2020).

El modelo BERT generalmente no se entrena desde cero, ya que su entrenamiento se ha realizado durante mucho tiempo con una gran cantidad de datos de Internet (Kobayashi et al. 2020). Para usarlo en diferentes aplicaciones de PLN, como el análisis de sentimientos, los investigadores generalmente agregan una o varias capas de redes neuronales sobre un modelo BERT previamente entrenado y entrenan la red para su propia tarea. Durante el entrenamiento, es posible bloquear los pesos del modelo BERT o ajustarlos con retropropagación (Kobayashi et al. 2020).

Antes de los transformadores de oraciones, el enfoque para calcular la similitud precisa de oraciones con BERT era usar una estructura de codificador cruzado. Esto significaba que se debería pasar dos oraciones a BERT, agregar un encabezado de clasificación en la parte superior de BERT y luego usar para generar una puntuación de similitud.

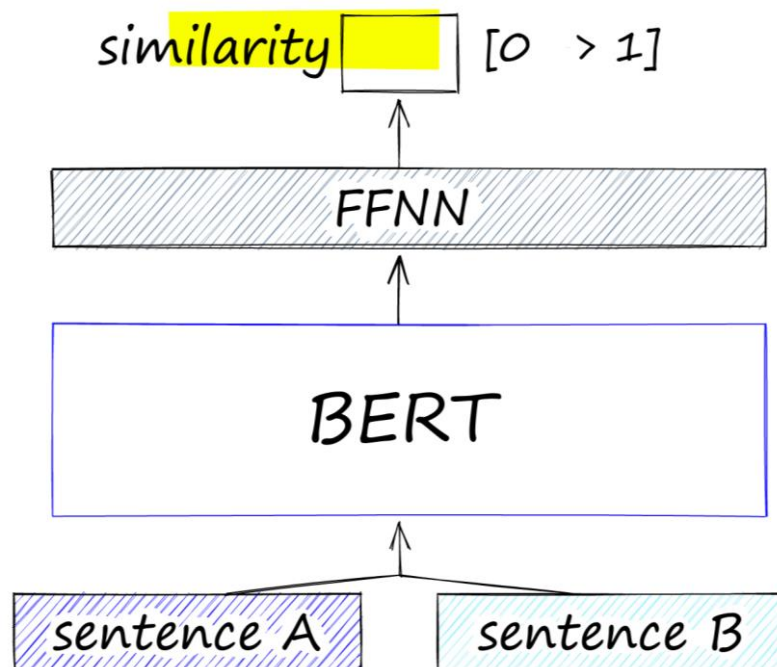


Figura 21. Arquitectura del codificador cruzado BERT (Briggs 2022)

Como se observa en la figura 21, la arquitectura de codificador cruzado BERT consta de un modelo BERT que consume las oraciones A y B. Ambas se procesan en la misma secuencia, separadas por un token [SEP]. Todo esto es seguido por el clasificador de red neuronal prealimentada (feedforward neural network en inglés) que genera una puntuación de similitud.

La red de codificadores cruzados produce puntajes de similitud muy precisos, pero no es escalable. Si se quiere realizar una búsqueda de similitud a través de un pequeño conjunto de datos de 100.000 oraciones, se necesita completar el cálculo de inferencia del codificador cruzado 100.000 veces. Con certeza no es un resultado para nada favorable.

La solución a esta falta de un modelo preciso con una latencia razonable fue diseñada por Nils Reimers e Iryna Gurevych en 2019 con la introducción de sentence-BERT (SBERT) y la biblioteca de sentence-transformers (transformadores de oraciones).

### Sentence BERT

BERT ha logrado un rendimiento de vanguardia en tareas de clasificación y regresión de oraciones, como la similitud semántica de dos oraciones. Sin embargo, es obligatorio proporcionar ambas oraciones a la red, lo que genera una costosa sobrecarga computacional. BERT tiene menor precisión para la búsqueda de similitud semántica y el agrupamiento no supervisado. El artículo Sentence-BERT (SBERT) propuesto por Reimers y Gurevych (Reimers, Gurevych 2019), es una adaptación del modelo BERT pre-entrenado que usa estructuras de red siamesa y triplete para producir embedding de oraciones semánticamente relevantes usando similitud de coseno para la comparación. Esta técnica reduce la sobrecarga para encontrar las oraciones más similares mientras mantiene la precisión de BERT.

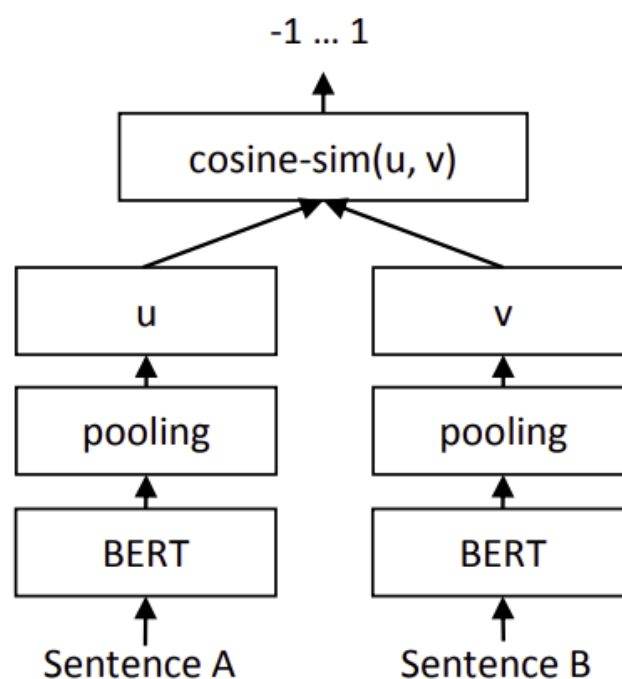


Figura 22. Arquitectura SBERT para calcular puntuaciones de similitud (Reimers, Gurevych 2019).

El modelo SBERT propuesto por (Reimers, Gurevych 2019) se utiliza en este trabajo para derivar la similitud semántica entre las descripciones de los pictogramas y una oración que se pretende traducir en pictogramas. El modelo ayuda a encontrar la descripción y/o el nombre en función del puntaje de similitud con relación a la oración/palabra de búsqueda de una manera más sencilla y con una ejecución más rápida que BERT. Como las descripciones están conectadas a los pictogramas, al devolver la descripción con mayor similitud semántica se estará devolviendo el pictograma asociado a dicha descripción.

A continuación se crean y codifican los embeddings de los nombres y descripciones con un modelo pre-entrenado (*hiiamsid/sentence\_similarity\_spanish\_es* · Hugging Face 2021) de SBERT y refinado con un conjunto de texto en español (Cañete 2022) facilitado por SentenceTransformers y disponible en la plataforma Hugging Face (*Hugging Face – The AI community building the future*. sin fecha).

### **2.4.3 Codificación y Creación de los Embeddings de los Nombres y de las Descripciones**

Para crear los embeddings de los nombres y de las descripciones se sigue el siguiente algoritmo: se obtienen los nombres y descripciones de los pictogramas desde ARASAAC, de manera individual se codifican los dos con el modelo SBERT, el resultado de esta codificación son los embeddings, una vez obtenidos los embeddings se indexan estos con las posiciones de los nombres/descripciones que estos embeddings representan, de este modo es posible más adelante recuperar el pictograma al que pertenece el embedding. Finalmente, se han creado dos modelos de embeddings indexados (uno para los nombres y otro para las descripciones). Esta indexación es clave en el mecanismo de búsqueda.

Como se mencionó anteriormente en el análisis lingüístico, los dos modelos son usados para fines distintos, en dependencia del etiquetado gramatical. Por ejemplo, las búsquedas de sustantivos o cadenas de texto con más de una palabra se realizan en los embeddings de las descripciones.

A continuación se muestran los resultados de la creación de los dos embeddings (Figura 23, 24).

## Embeddings de los Nombres

```
[6] # Encode Keyword
k_encode = model.encode(arasaac_mapping_db['keyword'].str.lower().tolist(), show_progress_bar=True)
k_encode

Batches: 100% ██████████ 559/559 [00:10<00:00, 82.77it/s]
array([[ 0.86772364, -0.2430551,  0.27330345, ..., -0.01846159,
        -0.8990736, -0.7640169 ],
       [ 1.0265658,  0.4593714, -1.0570168, ..., -0.5209167,
        -0.27533743, -0.10887001],
       [ 0.5088522,  1.2261666, -0.19846368, ..., -1.2228659,
        -0.2387032, -0.08565769],
       ...,
       [-0.6250743,  0.75360906, -1.080983, ...,  0.32423526,
        -0.15616664,  0.0977894 ],
       [-0.6565353,  0.46966043, -0.6067815, ...,  0.8686532,
        -0.40829223,  0.20357683],
       [ 0.80401933,  0.01328492, -0.5748879, ...,  0.72780854,
        -0.10395939, -0.1773985 ]], dtype=float32)
```

Figura 23. Embeddings de los nombres. Resultado de la codificación y creación de los embeddings de los nombres.

## Embeddings de las descripciones

```
[7] # Encode Meaning
m_encode = model.encode(arasaac_mapping_db['meaning'].str.lower().tolist(), show_progress_bar=True)
m_encode

Batches: 100% ██████████ 559/559 [00:35<00:00, 50.31it/s]
array([[ 0.13681653, -0.1057382,  0.4741342, ..., -0.44628263,
        -0.54604864, -0.17650549],
       [ 1.4479272,  0.5688215, -1.432388, ..., -0.35135105,
        -0.6089266,  0.1126956 ],
       [-0.12812139,  0.17529678, -0.35704446, ..., -0.7626816,
        -0.27934238, -0.50410056],
       ...,
       [-0.5466083,  0.74300694, -0.9040155, ...,  0.31226975,
        -0.4811556, -0.46972454],
       [-0.6040792,  0.44327244, -0.33039284, ...,  0.8857422,
        -0.4310628, -0.08636009],
       [ 1.0604513,  0.27611935, -1.1474214, ...,  1.0581932,
        0.3825231, -0.38509956]], dtype=float32)
```

Figura 24. Embeddings de las descripciones. Resultado de la codificación y creación de los embeddings de las descripciones.

## 2.5 Arquitectura del modelo creado

La arquitectura final del analizador semántico estará compuesta por un análisis lingüístico, un mecanismo de búsqueda semántica, una predicción en el modelo pre entrenado SBERT y una búsqueda en la API de ARASAAC.

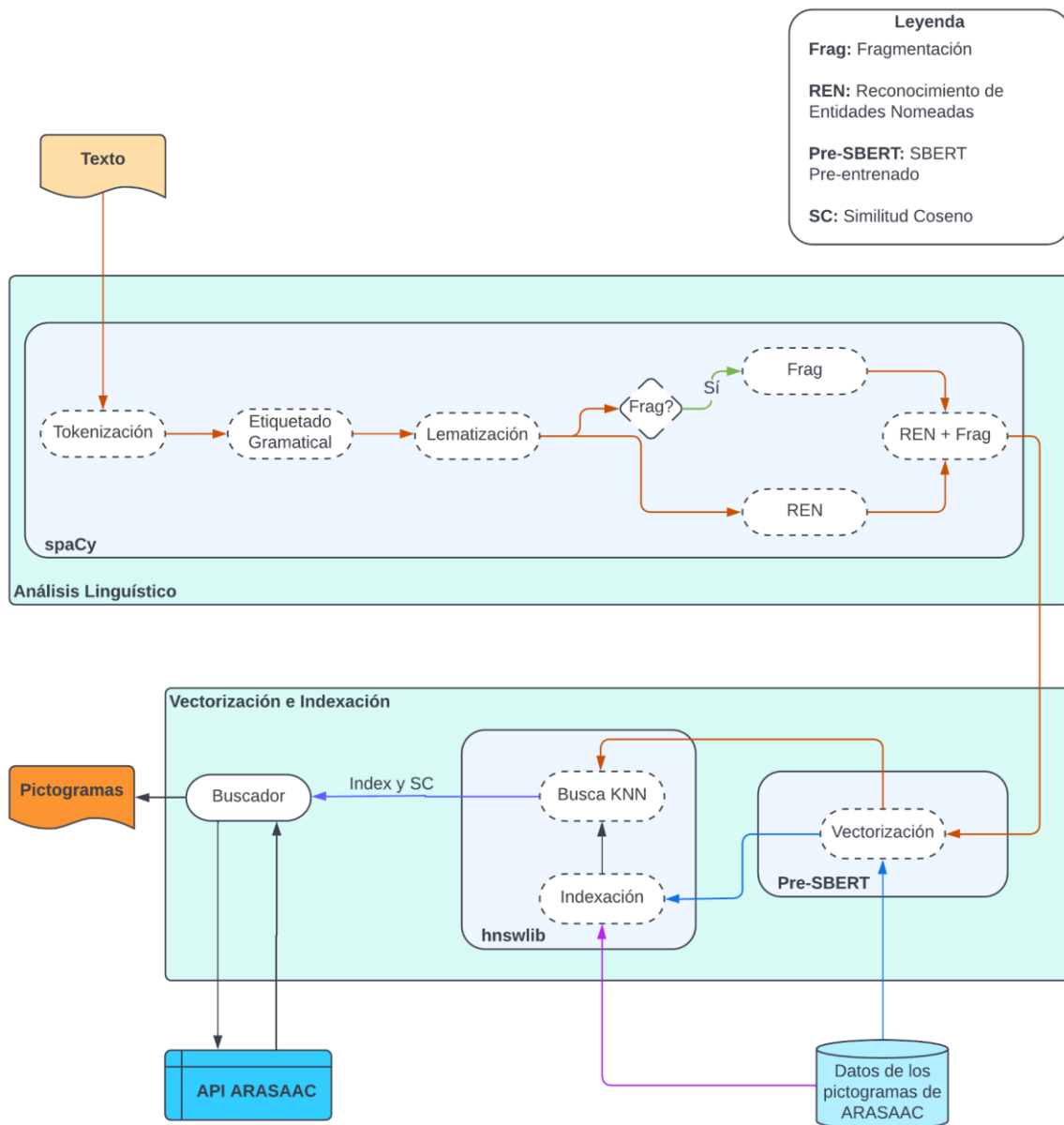


Figura 25. Arquitectura del analizador semántico

Cuando se realiza una búsqueda se requiere un proceso de representación, almacenamiento y organización de la información para conseguir el acceso y recuperación de información que responda a las necesidades de un usuario (Epifanio Tula, Medeot 2007). Este proceso de la información se ha realizado mediante Procesamiento del Lenguaje Natural, utilizando gran cantidad de información en formato de texto. La representación de palabras en un espacio vectorial es a través de un modelo denominado Word Embeddings. La Figura 25 muestra la arquitectura de la propuesta de solución.

Si se analiza la oración “Mamá está en la cocina”, primeramente se hace el análisis lingüístico formado por la tokenización, el etiquetado gramatical, la lematización y la fragmentación/ chunking. Esto permite más adelante mejorar la búsqueda y consecuentemente las traducciones. El objetivo final es traducir esta secuencia de palabras en secuencias de pictogramas, para eso se buscan los pictogramas usando los embeddings creados con el modelo SBERT, este mecanismo de búsqueda se llama “búsqueda por similitud semántica”, es decir, cuando por ejemplo se busca por la palabra mamá, se buscan todos los pictogramas que sus descripciones o nombres tienen una similitud semántica con la palabra mamá. Entonces se recorre palabra por palabra de la cadena de búsqueda con el objetivo de encontrar los pictogramas ideales. Los pictogramas ideales deben ser, primero semánticamente similar a cada palabra y segundo, apropiado dependiendo del contexto.

Cuando se busca un pictograma que represente una palabra, por ejemplo “Mamá”, primero se calcula el embedding de la cadena de búsqueda de la misma forma que se crearon los embeddings para los nombres y descripciones de los pictogramas, luego se comparan los embeddings de cada nombre o descripción con el embedding de la cadena de búsqueda, el embedding más cercano se pone en la primera posición y el embedding más distante en la última posición. Luego, los resultados son sustituidos por los respectivos pictogramas.

El algoritmo HNSW es una aproximación al algoritmo Vecino más cercano (KNN por sus siglas en inglés), pero con mayor rapidez y exactitud. Este es utilizado para hacer la búsqueda de la palabra clave o significado más cercano a la cadena de búsqueda que se desea representar.

Usando este enfoque de búsqueda de similitud semántica se garantiza una búsqueda eficiente de los pictogramas; sin embargo, los resultados aún dependen mucho de las descripciones que tengan los pictogramas, es decir, si las descripciones son pobres en cuanto al contexto y/o no describen correctamente el pictograma, entonces los resultados no serán eficientes desde el punto de vista humano. Se puede notar la importancia del etiquetado de los pictogramas.



## **2.6 Evaluación de la Solución Propuesta**

Para verificar que, con la implementación del analizador semántico, se aumenta la estabilidad de los pictogramas al contexto y a la semántica se emplea el método Delphi y el método científico cuantitativo de experimento descrito por Hernández Sampieri (Hernández, Fernández, Baptista 2016).

El diseño experimental consiste en manipular intencionalmente una acción para analizar sus posibles resultados. Según (Hernández, Fernández, Baptista 2016) un diseño experimental debe cumplir con tres requisitos: la manipulación intencional de una o más variables independientes, medición del efecto que la variable independiente tiene en la variable dependiente y control o la validez interna de la situación experimental.

Por otro lado, el método Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a un problema definido, se clasifica como uno de los métodos generales de prospectiva, que busca acercarse al consenso de un grupo de expertos con base en el análisis y la reflexión de dicho problema (García Valdés, Suárez Marín 2013). Para la aplicación del método se siguieron las siguientes etapas:

### **1. Elección de expertos**

Según las fuentes referenciadas, un experto es una persona, grupo de personas u organización con conocimientos amplios en un área particular del conocimiento, capaces de valorar, formular conclusiones objetivas y dar recomendaciones acerca del problema en cuestión.

En el caso de esta investigación, los expertos se seleccionaron teniendo en cuenta que cumplieran con los criterios siguientes:

- Especialista en enseñanza de personas con dificultad del habla.
- Profesor/a de la lengua española.
- Familiar de persona con dificultad del habla.
- Conocimientos de PLN.

Finalmente, se escogieron ocho expertos, considerando la presencia de un especialista en enseñanza de personas con dificultad del habla, dos profesores de la lengua española, cuatro familiares de persona con dificultad del habla y un estudiante con conocimientos de PLN.

Tabla 4. Datos de los expertos seleccionados.

Experto	Tarea	Tiempo en la tarea
1	Especialista en enseñanza de personas con dificultad del habla	Cinco años
2	Profesor de la lengua española	Cuatro años
3	Profesor de la lengua española	Tres años
4	Pariente de persona con dificultad del habla	Ocho años
5	Pariente de persona con dificultad del habla	Ocho años
6	Pariente de persona con dificultad del habla	Siete años
7	Pariente de persona con dificultad del habla	Siete años
8	Estudiante con conocimiento de PLN	Dos años

## 2. Preparación del instrumento

Una vez seleccionados los expertos se elaboró un documento con diez oraciones que fue sometido a la consideración de los expertos con el objetivo de validar los indicadores de la propuesta de solución. Por ser un experimento a base de hipótesis, las oraciones son traducidas por dos de los sistemas homólogos (Pictar y Pictotraductor) y por la propuesta de solución (acuñado Pictobana). Los expertos tienen la tarea de evaluar las traducciones de los tres sistemas.

La evaluación de una oración es efectuada de la siguiente manera:

**Primero, se evalúa cada token traducido en pictograma.** Se evalúa cada token de la oración procesada (que pasó por el análisis lingüístico) traducido en pictograma. Si el pictograma se ajusta al contexto de la oración obtiene un valor de 0.5, de lo contrario obtiene un valor de 0. Si el pictograma se ajusta semánticamente al token

traducido obtiene un valor de 0.5, de lo contrario obtiene un valor de 0. Para saber si un token fue correctamente traducido se suman los dos valores anteriores, el resultado de 0 indica que el token no cumple con ninguno de los indicadores, 0.5 indica que cumple con uno de los indicadores y 1 indica que cumple con los dos indicadores, lo que quiere decir que la traducción está libre de incoherencia y ambigüedad semántica. Este cálculo se hace para cada token traducido.

**Segundo, se calcula el porcentaje de acierto de traducción de la oración.** El porcentaje de acierto de traducción de una oración, se calcula mediante la fórmula:

$$P_i = \left( \sum_{i=1}^n T_i \right) * 100 / n$$

Donde:

P<sub>i</sub> = Porcentaje de acierto de traducción de la oración i.

T<sub>i</sub> = Traducción del token i.

n = Cantidad de tokens de la oración pos procesamiento (análisis lingüístico).

Los sistemas pictotraductor analizados y la solución propuesta, devuelven más de una ocurrencia para cada traducción de un token, con el objetivo de seleccionar el pictograma ideal si este no está en la primera ocurrencia. En la figura 26, se observa una traducción con 2 resultados. Por lo que se ha evaluado el porcentaje de acierto del primer resultado (primera fila) que devuelve el sistema, con el objetivo de analizar la efectividad del sistema al devolver la traducción correcta en el primer resultado, y también se evaluó el porcentaje de acierto dentro de los 5 primeros resultados.

La cantidad de tokens de la oración después del análisis lingüístico puede variar, siendo este valor menor o igual a la cantidad de tokens inicial de la oración. Como se observa en la figura 26, la oración inicial tiene cinco tokens y la traducción de Pictar y Pictobana consta de tres pictogramas representando tres tokens pertenecientes a las cadenas: “cierra el grifo”, “al” y “terminar”. Eso significa que Pictar y Pictobana realizaron la fragmentación. En este caso “n” es igual a tres. Para Pictotraductor la cantidad de tokens es igual a 5, la misma cantidad que la oración inicial (antes del análisis lingüístico).

—● **Oración:** Cierra el grifo al terminar.



Figura 26. Resultado de los pictotraductores ante la oración “Cierra el grifo al terminar”.

**Tercero, se calcula el promedio de acierto del sistema para las diez oraciones.**

El promedio de acierto del sistema, se calcula mediante la fórmula:

$$PA = \left( \sum_{i=1}^n Pi \right) / n$$

Donde:

PA = Promedio de acierto del sistema.

Pi = Porcentaje de acierto de traducción de la oración i.

n = Cantidad de oraciones.

En la tabla 5 se observan los resultados de la evaluación.

### **3. Promedio de las evaluaciones y concordancia de los expertos**

Los criterios de evaluación de las oraciones son aprobados por los expertos, por lo que hubo un consenso sobre el método a usar para la validación de las mismas. Por este motivo no hubo necesidad de determinar mediante cálculos el nivel de concordancia de las evaluaciones de los expertos. En la tabla 5 se muestra el resultado final de las evaluaciones.

### **4. Reporte de los resultados**

La tabla 5 muestra los resultados detallados para cada oración, las figuras 27 y 28 muestran los resultados finales de los sistemas homólogos y de la solución propuesta

para los resultados en la primera posición (figura 27) y en las cinco primeras posiciones (figura 28).

Como se evidencia en los resultados, Pictobana pudo hacer 5 fragmentaciones correctas (es decir que no distorsionan el significado de la oración), terminó con 85,94% de precisión para los resultados en la primera posición y 92,00% teniendo en cuenta las cinco primeras posiciones. Precisión superior a Pictar y Pictotraductor. Este resultado permite aceptar la hipótesis planteada inicialmente.

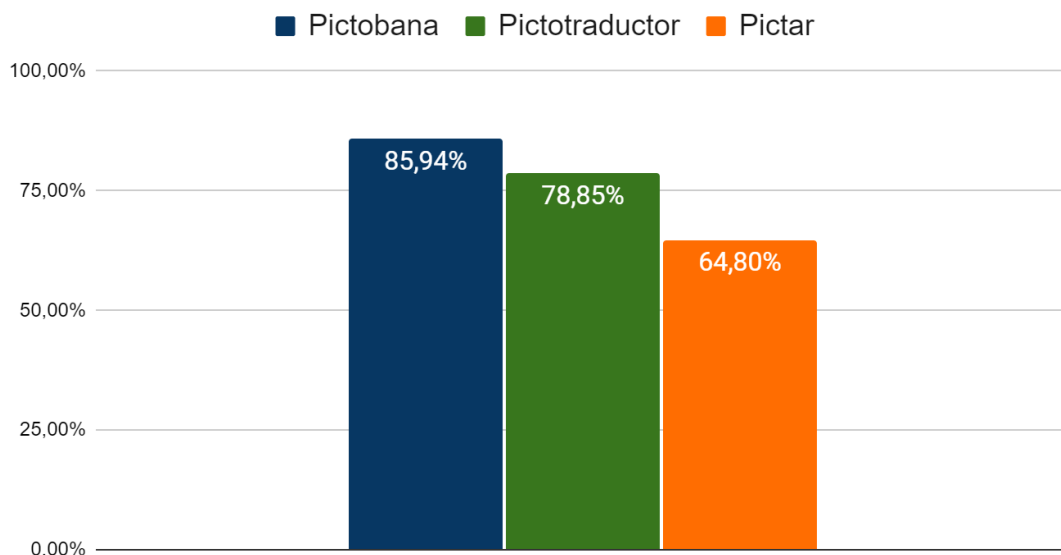


Figura 27. Resultados, porcentaje de acierto en la primera posición.

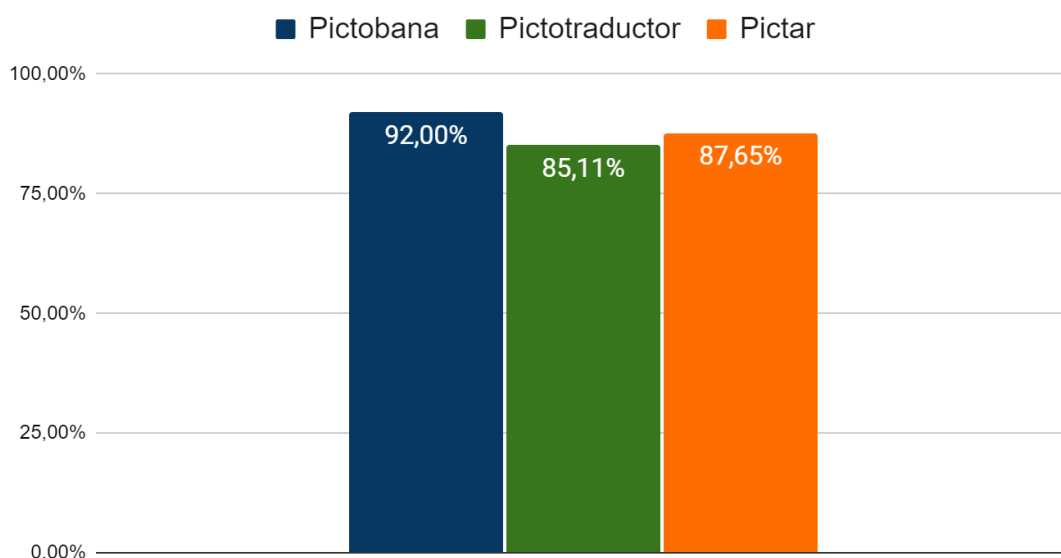


Figura 28. Resultados, porcentaje de acierto en las cinco primeras posiciones.

Tabla 5. Resultados de la evaluación de Pictobana en comparación a Pictar y Pictotraductor

Oración	Cant. Tokens	Frag/REN?; Correcta?			% de acierto 1r resultado			% de acierto 5 primeros resultados		
		Pictobana	Pictotraductor	Pictar	Pictobana	Pictotraductor	Pictar	Pictobana	Pictotraductor	Pictar
Cierra el grifo al terminar.	5	3tok; Sí	No	3tok; Sí	83,30%	70%	83,30%	100%	80%	100%
Lava tus palmas y dedos de las manos.	8	6tok; Sí	No	No	83,30%	75%	62,50%	100%	75%	87,50%
Enciende el televisor.	3	1tok; Sí	No	No	100%	83,30%	50%	100%	100%	100%
Escribe sin romper las hojas.	5	No	No	No	70%	70%	50%	80%	70%	80%
Yo me siento bien para descansar	6	No	No	No	100%	83,30%	83,30%	100%	83,30%	83,30%
El niño corre la cortina.	5	No	No	No	80%	80%	60%	80%	80%	80%
El conejo va a dormir.	5	No	No	No	70%	80%	60%	80%	90%	80%
El fuego devora la madera.	5	No	No	No	90%	80%	40%	90%	80%	80%
La niña tiene ojos cafe y pelo negro.	8	7tok; Sí	No	No	92,80%	81,20%	87,50%	100%	100%	100%
Papá corta el árbol con su hacha.	7	5tok; Sí	No	No	90%	85,70%	71,40%	90%	92,80%	85,70%
<b>Promedio</b>		<b>5 Frag</b>	<b>0 Frag</b>	<b>0 Frag</b>	<b>85,94%</b>	<b>78,85%</b>	<b>64,80%</b>	<b>92,00%</b>	<b>85,11%</b>	<b>87,65%</b>

## **2.7 Conclusiones Parciales**

El empleo de la metodología CRISP-DM, como guía para la implementación, permitió implementar un sistema que cumpliera con los estándares de desarrollo de proyectos de minería de datos y PLN. El análisis exploratorio de los datos recopilados, permitió identificar las características de los pictogramas más adecuadas para el analizador semántico y algunos inconvenientes en las características que disminuyen la eficacia de la solución propuesta. El analizador desarrollado proporciona a las personas con dificultades del habla en Cuba y a sus familiares una herramienta que ayuda a promover las competencias y habilidades de comunicación de estas personas. Finalmente, las pruebas realizadas mediante experimentos y criterio de expertos, validó que el sistema desarrollado, aumenta la estabilidad de los pictogramas al contexto y a la semántica, aminorando la incoherencia y la ambigüedad semántica del sistema.

## CONCLUSIONES

El desarrollo del analizador semántico para los Sistemas de Comunicación Aumentativa y Alternativa Pictotraductor permite arribar a las siguientes conclusiones:

- Con el estudio de los referentes teóricos sobre los sistemas de comunicación aumentativa y alternativa realizado en este capítulo se evidenció la importancia que tienen en el desarrollo de las personas con dificultad del habla, y del uso de un analizador semántico en estos sistemas.
- El análisis de los sistemas homólogos permitió detectar las funcionalidades básicas de un pictotraductor y del analizador semántico.
- El análisis de los sistemas homólogos demostró la necesidad de desarrollar un analizador semántico para los Sistemas de CAA Pictotraductor para facilitar a las personas con dificultades del habla en Cuba y a sus familiares una herramienta que les ayude a promover las competencias y habilidades de estas personas.
- El empleo de la metodología CRISP-DM, como guía para la implementación, permitió implementar un sistema que cumpliera con los estándares de desarrollo de proyectos de minería de datos y PLN.
- El análisis exploratorio de los datos recopilados, permitió identificar las características de los pictogramas más adecuadas para el analizador semántico y algunos inconvenientes en las características que disminuyen la eficacia de la solución propuesta.
- El sistema desarrollado proporciona a las personas con dificultades del habla en Cuba y a sus familiares una herramienta que ayuda a promover las competencias y habilidades de estas personas.
- Las pruebas realizadas mediante experimentos y criterio de expertos, validaron que el sistema desarrollado, aumenta la estabilidad de los pictogramas al contexto y a la semántica, aminorando la incoherencia y la ambigüedad semántica del sistema.



## RECOMENDACIONES

El tiempo limitado para el desarrollo del proyecto ha supuesto no poder implementar todas las funcionalidades e ideas deseadas que han ido surgiendo a lo largo del mismo. Para dar continuidad con la investigación descrito en este trabajo, a continuación, se exponen algunas modificaciones e ideas de cara a trabajos futuros:

- Revisar y editar las características de algunos pictogramas de tal manera que sea posible utilizar dicha información en apoyo al contexto de la oración.
- Traducir otras características de los pictogramas del dataset de ARASAAC al español y utilizarlas para reforzar la comprensión del pictograma.
- Almacenar las oraciones entradas por el usuario para usarlas como datos de entrenamiento del modelo.
- Realizar el análisis de dependencia para reforzar la comprensión de las relaciones de los elementos léxicos.
- Revisar y adaptar el dataset de ARASAAC al contexto cubano.

## REFERENCIAS BIBLIOGRÁFICAS

AAC INSTITUTE, 2022. AAC INSTITUTE » What is AAC? en línea. 2022. [Accedido 30 junio 2022]. Recuperado a partir de: <https://aacinstitute.org/what-is-aac/>

AAC Symbols and shared resources - ARASAAC, 2022. en línea. [Accedido 26 septiembre 2022]. Recuperado a partir de: <https://arasaac.org/>

ACUÑA, Jorge y MENDOZA, Carlos Invernizzi, 2020. Uso de Pictograma en paciente con Trastorno del Espectro Autista (TEA) en odontología. Relato de caso. *Revista Científica Odontológica*. 2020. Vol. 2, no. 1.

AGARWAL, Raunak, KONIAEV, Andrei y SCHAEFER, Robin, 2021. Exploring Argument Retrieval for Controversial Questions Using Retrieve and Re-rank Pipelines. En: *CLEF (Working Notes)*. 2021. pp. 2285-2291.

ALMEIDA, Felipe y XEXÉO, Geraldo, 2019. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*. 2019.

ANACONDA, sin fecha. Anaconda | The World's Most Popular Data Science Platform. en línea. [Accedido 18 septiembre 2022]. Recuperado a partir de: <https://www.anaconda.com/>

ARASAAC – CAREI, sin fecha. en línea. [Accedido 26 septiembre 2022]. Recuperado a partir de: <https://carei.es/descipcion-arasaac/>

ASHA, 2022a. Augmentative and Alternative Communication. *American Speech-Language-Hearing Association*. en línea. 2022. [Accedido 30 junio 2022]. Recuperado a partir de: <https://www.asha.org/practice-portal/professional-issues/augmentative-and-alternative-communication/>

ASHA, 2022b. Definition of Communication and Appropriate Targets. *American Speech-Language-Hearing Association*. en línea. 2022. [Accedido 30 junio 2022]. Recuperado a partir de: <https://www.asha.org/njc/definition-of-communication-and-appropriate-targets/>

BAHDANAU, Dzmitry, CHO, Kyunghyun y BENGIO, Yoshua, 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. 2014.

BARRANQUERO, Alejandro y BAEZA, Chiara Sáez, 2012. Teoría crítica de la comunicación alternativa para el cambio social. El legado de Paulo Freire y Antonio Gramsci en el diálogo Norte-Sur. *Razón y palabra*. 2012. No. 80.

BAUTISTA, Susana, HERVÁS, Raquel, HERNÁNDEZ-GIL, Agustín, MARTÍNEZ-DÍAZ, Carlos, PASCUA, Sergio y GERVÁS, Pablo, 2017. Aratrador: text to pictogram translation using natural language processing techniques. En: *Proceedings of the XVIII International Conference on Human Computer Interaction*. 2017. pp. 1-8.

BORRELLI, Dario, GONGORA SVARTZMAN, Gabriela y LIPIZZI, Carlo, 2020. Unsupervised acquisition of idiomatic units of symbolic natural language: An n-gram frequency-based approach for the chunking of news articles and tweets. *Plos one*. 2020. Vol. 15, no. 6, pp. e0234214.

BRIGGS, James, 2022. Sentence Transformers and Embeddings. *Pinecone*. en línea. 2022. [Accedido 5 noviembre 2022]. Recuperado a partir de: <https://www.pinecone.io/learn/sentence-embeddings/>

BROWNLEE, Jason, 2019. What Are Word Embeddings for Text? *Machine Learning Mastery*. en línea. 7 agosto 2019. [Accedido 23 octubre 2022]. Recuperado a partir de: <https://machinelearningmastery.com/what-are-word-embeddings/>

CABELLO, Laura, LLEIDA, Eduardo, SIMÓN, Javier, MIGUEL, Antonio y ORTEGA, Alfonso, 2018. Text-to-Pictogram Summarization for Augmentative and Alternative Communication. *Procesamiento del Lenguaje Natural*. 2018. Vol. 61, pp. 15-22.

CAÑETE, José, 2022. *Spanish Unannotated Corpora*. en línea. 20 agosto 2022. [Accedido 6 noviembre 2022]. Recuperado a partir de: <https://github.com/josecannete/spanish-corpora>

CHENG, Jianpeng, DONG, Li y LAPATA, Mirella, 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*. 2016.

CONTRERAS ARTEAGA, Abel Isaías y SÁNCHEZ COTRINA, Frank William, 2020. Analítica predictiva para conocer el patrón de consumo de los clientes en la Empresa Cienpharma SAC utilizando IBM SPSS Modeler y la metodología CRISP-DM. . 2020.

COOK, Albert M y POLGAR, Janice Miller, 2014. *Assistive technologies-e-book*:

*principles and practice*. Elsevier Health Sciences. ISBN 0-323-29101-5.

DESHMUKH, Rushali Dhumal y KIWELEKAR, Arvind, 2020. Deep learning techniques for part of speech tagging by natural language processing. En: *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE. 2020. pp. 76-81. ISBN 1-72814-167-2.

DEVLIN, Jacob, CHANG, Ming-Wei, LEE, Kenton y TOUTANOVA, Kristina, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. en línea. 11 octubre 2018. [Accedido 17 septiembre 2022]. DOI 10.48550/arXiv.1810.04805.

DONG, Li, YANG, Nan, WANG, Wenhui, WEI, Furu, LIU, Xiaodong, WANG, Yu, GAO, Jianfeng, ZHOU, Ming y HON, Hsiao-Wuen, 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*. 2019. Vol. 32.

ELCORREOGALLEGO, 2013. Comunicación gratuita para los autistas con el Grupo Promedia. [www.elcorreogallego.es](http://www.elcorreogallego.es). en línea. 13 febrero 2013. [Accedido 9 noviembre 2022]. Recuperado a partir de: <https://www.elcorreogallego.es/hemeroteca/comunicacion-gratuita-autistas-grupo-promedia-NKCG790962>

ELSAHAR, Yasmin, HU, Sijung, BOUAZZA-MAROUF, Kaddour, KERR, David y MANSOR, Annysa, 2019. Augmentative and alternative communication (AAC) advances: A review of configurations for individuals with a speech disability. *Sensors*. 2019. Vol. 19, no. 8, pp. 1911.

EPIFANIO TULA, Luis Geronimo y MEDEOT, Matías Daniel, 2007. Sistema de Recuperación de Información Sistema de Recuperación de Información Motor de Búsqueda: Innuendo. . 2007.

ESPINAL, M Teresa, MACIA, Josep, MATEU, Jaume y QUER, Josep, 2020. *Semántica*. Ediciones Akal. ISBN 84-460-4942-2.

FAULKNER, Autumn y CONTRIBUTOR, 2018. Lucidchart for easy workflow mapping. *Serials Review*. 2018. Vol. 44, no. 2, pp. 157-162.

FÉLIX-BRASDEFER, J César, 2018. *Pragmática del español: contexto, uso y variación*. Routledge. ISBN 1-315-44320-1.

FRANCO, Natália de Melo, 2020. Vocabulary selection and organization for augmentative and alternative communication of children with speech impairment. . 2020.

FRANCO, Natália, SILVA, Edson, LIMA, Rinaldo y FIDALGO, Robson, 2018. Towards a reference architecture for augmentative and alternative communication systems. En: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2018. pp. 1073. ISBN 2316-6533.

GARCÍA VALDÉS, Margarita y SUÁREZ MARÍN, Mario, 2013. El método Delphi para la consulta a expertos en la investigación científica. *Revista Cubana de Salud Pública*. 2013. Vol. 39, no. 2, pp. 253-267.

GELBUKH, Alexander, 2010. Procesamiento de lenguaje natural y sus aplicaciones. *Komputer Sapiens*. 2010. Vol. 1, pp. 6-11.

GOLDBERG, Yoav y LEVY, Omer, 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*. 2014.

GOOGLE, sin fecha. Welcome To Colaboratory - Colaboratory. en línea. [Accedido 18 septiembre 2022]. Recuperado a partir de: <https://colab.research.google.com/>

Grupo-Promedia, sin fecha. en línea. [Accedido 9 noviembre 2022]. Recuperado a partir de: <https://www.grupopromedia.es/>

HERNÁNDEZ, R, FERNÁNDEZ, C y BAPTISTA, P, 2016. Metodología de la investigación. 6ta Edición Sampieri. *Soriano, RR (1991). Guía para realizar investigaciones sociales. Plaza y Valdés*. 2016.

hiiamsid/sentence\_similarity\_spanish\_es · Hugging Face, 2021. en línea. [Accedido 25 octubre 2022]. Recuperado a partir de: [https://huggingface.co/hiiamsid/sentence\\_similarity\\_spanish\\_es](https://huggingface.co/hiiamsid/sentence_similarity_spanish_es)

Hugging Face – The AI community building the future., sin fecha. en línea. [Accedido 25 octubre 2022]. Recuperado a partir de: <https://huggingface.co/>

IBM, CRISP-DM, 2021. IBM Documentation. en línea. 17 agosto 2021. [Accedido 19 septiembre 2022]. Recuperado a partir de: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>

IBM SPSS Modeler CRISP-DM Guide, 2022. . pp. 56.

INTEL, Graphics processing technology has evolved to deliver unique benefits in the world of computing The latest graphics processing unitsunlock new possibilities in, CREATION, Content, LEARNING, Machine, y MORE., 2022. What Is a GPU? Graphics Processing Units Defined. *Intel*. en línea. 2022. [Accedido 26 octubre 2022]. Recuperado a partir de: <https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html>

KAHRAMONOVNA, Maxmudjanova Diana, 2021. LINGUISTIC ANALYSIS. *Web of Scientist: International Scientific Research Journal*. 2021. Vol. 2, no. 07, pp. 158-162.

KHATTAK, Faiza Khan, JEBLEE, Serena, POU-PROM, Chloé, ABDALLA, Mohamed, MEANEY, Christopher y RUDZICZ, Frank, 2019. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*. 2019. Vol. 100, pp. 100057.

KHEKARE, Ganesh y VERMA, Pushpneel, 2020. Design of Automatic Key Finder for Search Engine Optimization in Internet of Everything. En: *2020 IEEE 1st International Conference for Convergence in Engineering (ICCE)*. IEEE. 2020. pp. 464-468. ISBN 1-72817-340-X.

KHYANI, Divya, SIDDHARTHA, BS, NIVEDITHA, NM y DIVYA, BM, 2021. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology*. 2021.

KOBAYASHI, Goro, KURIBAYASHI, Tatsuki, YOKOI, Sho y INUI, Kentaro, 2020. Attention is not only a weight: Analyzing transformers with vector norms. *arXiv preprint arXiv:2004.10102*. 2020.

LIM, Kyungtae y PARK, Jungyeul, 2020. Part-of-speech tagging using multiview learning. *IEEE Access*. 2020. Vol. 8, pp. 195184-195196.

MALKOV, Yu A y YASHUNIN, Dmitry A, 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*. 2018. Vol. 42, no. 4, pp. 824-836.

MANDUJANO, Karla Patricia López, AGUILAR, Evelyn Anahy Pérez, BAUTISTA, Johana Virginia Ríos y ORTÍZ, Elia Haydée Carrasco, 2021. Uso del contexto semántico en el reconocimiento visual de palabras en niños lectores. *Estudios de Lingüística Aplicada*. 2021. No. 72, pp. 49-67.

MARÍN MARTÍNEZ, Antonia, 2019. Sistemas Alternativos y Aumentativos de Comunicación. *PublicacionesDidacticas.com*. 2019. Vol. 104, pp. 235-240.

MARTÍN, Alejandro, HERVÁS, Raquel, MÉNDEZ, Gonzalo y BAUTISTA, Susana, 2018. PICTAR: Una herramienta de elaboración de contenido para personas con TEA basada en la traducción de texto a pictogramas. *XIX Interacción*. 2018.

MAULUD, Dastan Hussen, ZEEBAREE, Subhi RM, JACKSI, Karwan, SADEEQ, Mohammed A Mohammed y SHARIF, Karzan Hussein, 2021. State of art for semantic analysis of natural language processing. *Qubahan Academic Journal*. 2021. Vol. 1, no. 2, pp. 21-28.

MIKOLOV, Tomas, CHEN, Kai, CORRADO, Greg y DEAN, Jeffrey, 2013. *Efficient Estimation of Word Representations in Vector Space*. en línea. 6 septiembre 2013. arXiv. arXiv:1301.3781. [Accedido 17 septiembre 2022]. Recuperado a partir de: <http://arxiv.org/abs/1301.3781> arXiv:1301.3781 [cs]

MIKOLOV, Tomas, CHEN, Kai, CORRADO, Gregory S. y DEAN, Jeffrey A., 2017. Computing numeric representations of words in a high-dimensional space en línea. US9740680B1. 22 agosto 2017. [Accedido 24 octubre 2022]. Recuperado a partir de: <https://patents.google.com/patent/US9740680B1/en>US14715421

NAKAMURA, Carlos y ZENG-TREITLER, Qing, 2012. A taxonomy of representation strategies in iconic communication. *International journal of human-computer studies*. 2012. Vol. 70, no. 8, pp. 535-551.

NIVRE, Joakim, HALL, Johan, NILSSON, Jens, CHANEV, Atanas, ERYIGIT, Gülşen, KÜBLER, Sandra, MARINOV, Svetoslav y MARSI, Erwin, 2007. MaltParser: A

language-independent system for data-driven dependency parsing. *Natural Language Engineering*. 2007. Vol. 13, no. 2, pp. 95-135.

pandas documentation, 2022. en línea. [Accedido 22 octubre 2022]. Recuperado a partir de: <https://pandas.pydata.org/docs/>

PARK, Kyubyong, LEE, JooHong, JANG, Seongbo y JUNG, Dawoon, 2020. An empirical study of tokenization strategies for various Korean NLP tasks. *arXiv preprint arXiv:2010.02534*. 2020.

PENNINGTON, Jeffrey, SOCHER, Richard y MANNING, Christopher, 2014. Glove: Global Vectors for Word Representation. En: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. en línea. Doha, Qatar: Association for Computational Linguistics. 2014. pp. 1532-1543. [Accedido 17 septiembre 2022]. DOI 10.3115/v1/D14-1162.

PEREIRA, Jayr Alencar, 2020. Uma gramática semântica baseada em Colorful Semantics para Comunicação Aumentativa e Alternativa. . 2020.

PEREIRA, Jayr Alencar, MACÊDO, David, ZANCHETTIN, Cleber, DE OLIVEIRA, Adriano Lorena Inácio y DO NASCIMENTO FIDALGO, Robson, 2022. PictoBERT: Transformers for next pictogram prediction. *Expert Systems with Applications*. 2022. Vol. 202, pp. 117231.

PICTAR, sin fecha. en línea. [Accedido 28 octubre 2022]. Recuperado a partir de: <http://hypatia.fdi.ucm.es/pictar/>

Pictoagenda, sin fecha. en línea. [Accedido 6 noviembre 2022]. Recuperado a partir de: <https://www.pictoagenda.com/>

PictoNews :: Creando-y-educando, 2018. en línea. [Accedido 6 noviembre 2022]. Recuperado a partir de: <https://creando-y-educando.webnode.es/pictonews/>

Pictotraductor, 2013. en línea. [Accedido 12 octubre 2022]. Recuperado a partir de: <https://www.pictotraductor.com/>

PRADHAN, Nitesh, COMPUTER, M. Tech Scholar, PHD, Manasi Gyanchandani y PHD, Rajesh Wadhvani, sin fecha. Raytheon BBN Technologies. En: *2011 (04). CoNLL-2011 Shared Task - Task Description*. <http://conll.bbn.com/index.php/task->



*description.html*.

PYTHON.ORG, sin fecha. Welcome to Python.org. *Python.org*. en línea. [Accedido 26 septiembre 2022]. Recuperado a partir de: <https://www.python.org/>

QIU, Xipeng, SUN, Tianxiang, XU, Yige, SHAO, Yunfan, DAI, Ning y HUANG, Xuanjing, 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*. 2020. Vol. 63, no. 10, pp. 1872-1897.

QU, Chen, YANG, Liu, QIU, Minghui, CROFT, W Bruce, ZHANG, Yongfeng y IYYER, Mohit, 2019. BERT with history answer embedding for conversational question answering. En: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019. pp. 1133-1136.

RAE, 2022. RAE. *Real Academia Española*. en línea. 2022. [Accedido 7 octubre 2022]. Recuperado a partir de: <https://www.rae.es/inicio>

REIMERS, Nils y GUREVYCH, Iryna, 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*. 2019.

RODRÍGUEZ, O, 2010. Metodología para el desarrollo de proyectos en minería de datos CRISP-DM. . 2010.

RODRÍGUEZ VELECELA, María Belén y JUELA LEÓN, Sonia Elizabeth, 2021. Uso de sistemas alternativos y aumentativos de comunicación y su incidencia en el desarrollo del lenguaje y comunicación de niños y niñas con discapacidad motriz. . 2021.

ROY, Subham, MODAK, Akshay, BARIK, Debabrata y GOON, Surajit, 2019. An overview of semantic search engines. *Int. J. Res. Rev.* 2019. Vol. 6, no. 10, pp. 73-85.

RUEDA, Ricardo Adán Salas y ESTUPIÑÁN, José de Jesús Vázquez, 2018. Aplicación en la nube Lucidchart: ¿herramienta necesaria para la innovación del proceso educativo en el siglo XXI? *Revista de Comunicación de la SEECI*. 2018. No. 44, pp. 115-126.

SentenceTransformers Documentation, 2022. en línea. [Accedido 22 octubre 2022]. Recuperado a partir de: <https://www.sbert.net/>

SONG, Youwei, WANG, Jiahai, LIANG, Zhiwei, LIU, Zhiyue y JIANG, Tao, 2020. Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference. *arXiv preprint arXiv:2002.04815*. 2020.

SPACY, sin fecha. spaCy · Industrial-strength Natural Language Processing in Python. en línea. [Accedido 18 septiembre 2022]. Recuperado a partir de: <https://spacy.io/>

SUTSKEVER, Ilya, VINYALS, Oriol y LE, Quoc V, 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*. 2014. Vol. 27.

teapps, 2022. en línea. [Accedido 6 noviembre 2022]. Recuperado a partir de: <https://aulautista.wixsite.com/teapps>

TÖNSING, Kerstin M., VAN NIEKERK, Karin, SCHLÜNZ, Georg y WILKEN, Ilana, 2019. Multilingualism and augmentative and alternative communication in South Africa – Exploring the views of persons with complex communication needs. *African Journal of Disability*. en línea. 24 abril 2019. Vol. 8. [Accedido 10 noviembre 2022]. DOI 10.4102/ajod.v8i0.507.

TOUKOUMIDIS, Ángel Torres, RODRÍGUEZ, Luis Miguel Romero y RODRÍGUEZ, María Amor Pérez, 2018. Ludificación y sus posibilidades en el entorno de blended learning: revisión documental. *RIED: revista iberoamericana de educación a distancia*. 2018. Vol. 21, no. 1, pp. 95-111.

Universal POS tags, 2022. en línea. [Accedido 29 octubre 2022]. Recuperado a partir de: <https://universaldependencies.org/u/pos/all.html>

UPPAL, Nitika, RANA, Shivani y SHARMA, Avni, 2019. Semantic web mining and semantic search engine: A review. . 2019.

VASWANI, Ashish, SHAZEER, Noam, PARMAR, Niki, USZKOREIT, Jakob, JONES, Llion, GOMEZ, Aidan N, KAISER, Łukasz y POLOSUKHIN, Illia, 2017. Attention is all you need. *Advances in neural information processing systems*. 2017. Vol. 30.

YADAV, Vikas y BETHARD, Steven, 2019. *A Survey on Recent Advances in Named Entity Recognition from Deep Learning models*. en línea. 24 octubre 2019. arXiv.

arXiv:1910.11470. [Accedido 18 septiembre 2022]. Recuperado a partir de:  
<http://arxiv.org/abs/1910.11470>arXiv:1910.11470 [cs]

ZORRILLA VILLANUEVA, Andrés, 2018. Búsqueda de imágenes basada en el Procesamiento de Lenguaje Natural aplicada en una Agenda Visual orientada a Personas con Trastorno del Espectro Autista. . 2018.