

INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Ingeniería Mecánica y Eléctrica

*Ingeniería en Comunicaciones y Electrónica
Especialidad en Computación*

**Tópicos Selectos de Ingeniería
Inteligencia Artificial**

***“Implementación de un Traductor Automático Inglés-
Español/Español-Inglés Mediante Memorias Asociativas
Morfológicas”***

Presentadores

Ambriz Martínez Ángel
Guajardo Guerra Bernardo Augusto

Asesor Técnico

Dra. María Elena Acevedo Mosqueda
Marco Antonio Acevedo Mosqueda

Asesor Metodológico

MC Beatriz A. Jaime Fonseca



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”

TEMA DE TESIS

**QUE PARA OBTENER EL TÍTULO DE
POR LA OPCIÓN DE TITULACIÓN
DEBERA (N) DESARROLLAR**

**INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL
C. BERNARDO AUGUSTO GUAJARDO GUERRA
C. ANGEL AMBRIZ MARTINEZ**

**“IMPLEMENTACIÓN DE UN TRADUCTOR AUTOMÁTICO INGLÉS-ESPAÑOL/ESPAÑOL-INGLÉS
MEDIANTE MEMORIAS ASOCIATIVAS MORFOLÓGICAS.”**

COMPROBAR QUE USANDO MEMORIAS ASOCIATIVAS BIDIRECCIONALES MORFOLÓGICAS, ES POSIBLE PROGRAMAR UN DICCIONARIO BILINGÜE EFICIENTE, ES DECIR CON UN TIEMPO DE CARGA Y RESPUESTA; ADEMÁS DE TOLERANCIA AL ERROR POR PARTE DEL USUARIO DEL SISTEMA.

- INTRODUCCIÓN
- ESTADO DEL ARTE
- MARCO TEÓRICO
- CONSTRUCCIÓN DEL PROYECTO
- PRUEBAS Y RESULTADOS
- CONCLUSIONES
- REFERENCIAS
- ANEXO
- SIMBOLOGÍA

MÉXICO D.F. A 15 DE DICIEMBRE DE 2015

ASESORES

M. EN C. MARÍA ELENA ACEVEDO MOSQUEDA

M. EN C. MARCO ANTONIO ACEVEDO MOSQUEDA

ING. BEATRIZ ADRIANA JAIME FONSECA

ING. PATRICIA LORENA RAMÍREZ RANGEL
JEFE DEL DEPARTAMENTO DE
INGENIERÍA EN COMUNICACIONES Y ELECTRONICA

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	7
1.1. PLANTEAMIENTO DEL PROBLEMA	7
1.2. JUSTIFICACIÓN.....	7
1.3. HIPÓTESIS.....	8
1.4. OBJETIVOS	8
a) OBJETIVO GENERAL.....	8
b) OBJETIVOS ESPECÍFICOS.....	9
1.5. ALCANCE DEL PROYECTO.....	9
1.6. ESTRUCTURA DEL PROYECTO	9
CAPÍTULO 2. ESTADO DEL ARTE	10
2.1. TRADUCCIÓN AUTOMÁTICA.....	10
2.2. TIPOS DE TRADUCCIÓN AUTOMÁTICA	11
2.2.1. TRADUCCIÓN AUTOMÁTICA BASADA EN REGLAS.....	11
2.2.2. TRADUCCIÓN AUTOMÁTICA BASADA EN CORPUS.....	12
2.3. EL TRADUCTOR MÁS UTILIZADO EN LA ACTUALIDAD: GOOGLE TRANSLATE ..	14
2.3.1. CARACTERÍSTICAS Y LIMITACIONES	14
2.4. MEMORIAS DE TRADUCCIÓN (MT)	16
2.4.1. FUNCIONAMIENTO DE UN SISTEMA DE MT	18
2.4.2. CREACIÓN DE UNA MT.....	18
2.4.3. TRABAJO CON UNA MT EXISTENTE	22
2.4.4. MEMORIAS DE TRADUCCIÓN COMERCIALES.....	23
CAPITULO 3. MARCO TEÓRICO	25
3.1. MEMORIAS ASOCIATIVAS	25
3.2. MEMORIAS LINEAR ASSOCIATOR	28
3.2.1. FASE DE APRENDIZAJE	29
3.2.2. FASE DE RECUPERACIÓN.....	29
3.2.3. EJEMPLO DE LINEAR ASSOCIATOR.....	30
3.3. MEMORIAS ASOCIATIVAS MORFOLÓGICAS.....	32

3.3.1.	FASE DE APRENDIZAJE	33
3.3.2.	FASE DE RECUPERACIÓN.....	34
3.3.3.	EJEMPLO DE UNA MEMORIA ASOCIATIVA MORFOLÓGICA.....	34
3.4.	VECTOR ONE HOT	38
3.5.	VECTORES ZERO HOT	39
3.6.	MEMORIAS BAM	40
CAPÍTULO 4. CONSTRUCCIÓN DEL PROYECTO		41
4.1.	DESCRIPCIÓN DE LAS MEMORIAS ASOCIATIVAS BIDIRECCIONALES MORFOLÓGICAS	41
4.1.1.	FASE DE APRENDIZAJE	41
4.1.2.	FASE DE RECUPERACIÓN.....	45
4.2.	DESARROLLO DEL PROTOTIPO	50
4.2.1	DIAGRAMA GENERAL DEL SISTEMA: DICCIONARIO BILINGÜE.....	51
4.2.2	FASES DE APRENDIZAJE.....	54
4.2.3	FASES DE RECUPERACIÓN	67
4.3.	EJEMPLO BAM MORFOLÓGICA.....	77
4.3.1.	FASE DE APRENDIZAJE	77
4.3.2.	FASE DE RECUPERACIÓN.....	79
CAPÍTULO 5. PRUEBAS Y RESULTADOS		82
5.1.	PRESENTACIÓN DEL PROTOTIPO	82
5.2.	PRUEBAS SOBRE EL PROTOTIPO	86
5.2.1	CONSIDERACIONES PREVIAS.....	86
5.2.2	CAPACIDAD DE RECUPERACIÓN.....	87
5.2.3	TIEMPOS DE CARGA Y RESPUESTA.....	87
5.2.4	TOLERANCIA AL RUIDO.....	88
5.3.	CASOS DE USO	88
5.4.	CONCLUSIONES	88
5.4.1	CONCLUSIONES EN BASE A LOS OBJETIVOS ESPECÍFICOS	89
REFERENCIAS		90
ANEXO		91
EJEMPLO NUMERICO DEL ALGORITMO.....		91
SIMBOLOGÍA.....		101

INDICE DE FIGURAS

FIGURA 1: ESQUEMA QUE MUESTRA LA RELACIÓN ENTRE LOS DIFERENTES PARADIGMAS DE TRADUCCIÓN AUTOMÁTICA BASADA EN REGLAS.....	11
FIGURA 2: ESQUEMA GENERAL DE UNA MEMORIA ASOCIATIVA BIDIRECCIONAL.....	40
FIGURA 3: DIAGRAMA GENERAL QUE MUESTRA EL FUNCIONAMIENTO DE LA INTERFAZ DE USUARIO.....	52
FIGURA 4: SEGUNDA PARTE DEL DIAGRAMA ANTERIOR.....	53
FIGURA 5: TERCERA PARTE DEL DIAGRAMA.....	54
FIGURA 6: PRIMERA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE DEL DICCIONARIO.....	55
FIGURA 7: SEGUNDA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE DEL DICCIONARIO.....	56
FIGURA 8: PRIMERA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE DE BAM MORFOLÓGICA.....	57
FIGURA 9: SEGUNDA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE DE BAM MORFOLÓGICA.....	58
FIGURA 10: DIAGRAMA DE LA FASE DE APRENDIZAJE LINEAR.....	59
FIGURA 11: PRIMERA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE MORFOLÓGICA.....	60
FIGURA 12: SEGUNDA PARTE DEL DIAGRAMA DE LA FASE DE APRENDIZAJE MORFOLÓGICA.....	61
FIGURA 13: PRIMERA PARTE DEL OPERADOR RESTA.....	62
FIGURA 14: SEGUNDA PARTE DEL OPERADOR RESTA.....	63
FIGURA 15: PRIMERA PARTE DEL OPERADOR MÁXIMO.....	64
FIGURA 16: SEGUNDA PARTE DEL OPERADOR MÁXIMO.....	65
FIGURA 17: PRIMERA PARTE DEL OPERADOR MÍNIMO.....	66
FIGURA 18: SEGUNDA PARTE DEL OPERADOR MÍNIMO.....	67
FIGURA 19: PRIMERA PARTE DEL DIAGRAMA DE RECUPERACIÓN DEL DICCIONARIO.....	68
FIGURA 20: SEGUNDA PARTE DEL DIAGRAMA DE RECUPERACIÓN DEL DICCIONARIO.....	69
FIGURA 21: PRIMERA PARTE DEL DIAGRAMA DE LA FASE DE RECUPERACIÓN DE BAM MORFOLÓGICA.....	71
FIGURA 22: SEGUNDA PARTE DEL DIAGRAMA DE LA FASE DE RECUPERACIÓN DE BAM MORFOLÓGICA.....	72
FIGURA 23: TERCERA PARTE DEL DIAGRAMA DE LA FASE DE RECUPERACIÓN DE BAM MORFOLÓGICA.....	73
FIGURA 24: DIAGRAMA DE LA FASE DE RECUPERACIÓN MORFOLÓGICA.....	74
FIGURA 25: PRIMERA PARTE DEL DIAGRAMA DE LA FUNCIÓN DE OPERACIÓN MORFOLÓGICA DE DILATACIÓN.....	75
FIGURA 26: SEGUNDA PARTE DEL DIAGRAMA DE LA FUNCIÓN DE OPERACIÓN MORFOLÓGICA DE DILATACIÓN.....	76
FIGURA 27: TERCERA PARTE DEL DIAGRAMA DE LA FUNCIÓN DE OPERACIÓN MORFOLÓGICA DE DILATACIÓN.....	77
FIGURA 28: PANTALLA INICIAL. SE MUESTRAN LOS NOMBRES DE LOS AUTORES.....	82
FIGURA 29: PANTALLA PARA LA SELECCIÓN DE DICCIONARIO INICIAL.....	83
FIGURA 30: MENÚ PRINCIPAL PARA TRADUCIR DE ESPAÑOL A INGLÉS.....	83
FIGURA 31: EJEMPLO DE UNA TRADUCCIÓN DE ESPAÑOL-INGLÉS.....	84
FIGURA 32: MENÚ DE OPCIONES PARA CAMBIAR DE TIPO DE DICCIONARIO.....	85
FIGURA 33: ES POSIBLE DESPLEGAR LA BASE DE DATOS COMPLETA.....	85

RESUMEN

En el capítulo 1 se plantea el problema a tratar, en la sección 1.2 se realiza la justificación seguida de una hipótesis; objetivo general que habla en cuestión de la meta propuesta para este proyecto, después se mencionan los objetivos específicos, los cuales son pequeños requisitos para cumplir con el fin principal.

En la sección 1.5 se tratan los alcances del proyecto y en la sección final se da una breve explicación de cómo está estructurado el mismo.

Ya para el capítulo dos entramos al estado del arte, donde mencionamos breves artículos en los que se han implementado diversas Memorias de traducción, de igual manera explicamos diferentes procesos de traducción y damos información acerca de cómo funciona el traductor más utilizado en el mundo.

En el capítulo 3 conocido como marco teórico se profundiza en los métodos algebraicos más ocupados por las memorias asociativas, ponemos al alcance del lector todos los medios necesarios para entender y comprender el algoritmo propuesto en el capítulo 4.

En dicho capítulo implementamos un nuevo algoritmo llamado Memorias BAM Morfológicas. Aquí lo dividimos en 3 secciones la sección 4.1 que es toda la explicación teórica y matemática. En la 4.2 se hace referencia al software prototipo, explicándolo con diagramas de flujo y en la sección 4.3 un ejemplo numérico del algoritmo.

Para finalizar está el capítulo 5, en este se plasman los resultados de las pruebas hechas al software, los problemas que se presentaron y las conclusiones según los objetivos.

CAPÍTULO 1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Algunos traductores de idiomas automáticos (específicamente los que hacen uso de Memorias de Traducción) presentan una dificultad en su funcionamiento: acceder de forma eficaz a una base de datos que alberga traducciones hechas previamente.

Los algoritmos de memorias asociativas básicos no están diseñados para el manejo de dichas bases de datos, ya que trabajan de forma óptima con valores binarios, y no así con valores de otros sistemas numéricos.

El presente proyecto pretende comprobar que es posible aplicar un nuevo algoritmo de memorias asociativas al registro de traducciones.

1.2. JUSTIFICACIÓN

Los algoritmos de traducción automática actuales son más eficientes cuando conservan un registro de traducciones previas hechas por el mismo algoritmo, ya que evitan gastar tiempo y recursos en traducir más de una vez una misma palabra o frase. Sin embargo, acceder a estos “registros” por métodos convencionales (como leer el archivo de principio a fin hasta encontrar una coincidencia) puede resultar contraproducente.

El problema con las memorias asociativas se agrava cuando introduces valores no binarios, teniendo en cuenta que cada carácter equivale a un valor ASCII, y que en una base de datos existen muchos caracteres.

Aunque ya hay soluciones diversas a este problema, es muy importante que este proceso sea lo más eficiente posible, para asegurar la pronta respuesta del traductor.

1.3. HIPÓTESIS

Las memorias asociativas se prestan como una alternativa para acceder a archivos que funcionan como bases de datos de palabras ya traducidas. A grandes rasgos, su funcionamiento consiste en asociar una lista de palabras en un idioma, con otra lista que contenga las traducciones de las palabras en otro idioma.

Existen varios algoritmos y métodos para crear memorias asociativas, cada uno con sus ventajas y debilidades.

Usar Memorias Asociativas Bidireccionales (BAM) combinadas con memorias Alfa Beta, puede hacer que estas últimas sean tolerantes ante ruido mezclado, ya que por sí solas las memorias Alfa Beta solo pueden soportar ruido aditivo o sustractivo.

Lo mismo pasa si se combinan las Bidireccionales con memorias Morfológicas, pero además con la ventaja de que las memorias morfológicas pueden manejar datos enteros (a diferencia de las Alfa Beta que solo manejan datos binarios). Esta es la razón por la que se decidió trabajar con esta combinación de memorias, ya que un texto electrónico se representa como valores enteros ASCII por cada letra; el resultado es que se generarían memorias asociativas más eficientes y pequeñas.

1.4. OBJETIVOS

a) OBJETIVO GENERAL

El objetivo de este proyecto es comprobar que usando Memorias Asociativas Bidireccionales Morfológicas, es posible programar un diccionario bilingüe eficiente, es decir, con un tiempo de carga y de respuesta corto; y que además soporte errores de escritura por parte del usuario del sistema.

b) OBJETIVOS ESPECÍFICOS

- A partir del algoritmo BAM Alfa-Beta desarrollar de forma análoga el algoritmo BAM Morfológica e implementarlo en un programa en Matlab.
- Modificar el código para que funcione con palabras en vez de números.
- Adaptar el código para que pueda acceder a bases de datos de palabras en inglés y español.
- Determinar cuál es la cantidad adecuada de palabras que deben llevar las bases de datos para que el software funcione mejor.
- Encontrar la tolerancia al error que soporta el sistema por parte del usuario.

1.5. ALCANCE DEL PROYECTO

El presente proyecto pretende comprobar qué tan eficiente es trabajar con memorias BAM morfológicas, por lo que sólo se pretende programar un prototipo de traductor para dos idiomas.

1.6. ESTRUCTURA DEL PROYECTO

El prototipo del programa traductor deberá estar conformado por los siguientes elementos:

- Capa de presentación para interactuar con el usuario.
- Capa de bases de datos, con una lista de palabras en inglés y otra en español.
- Capa del algoritmo BAM Morfológica, dividido en fase de aprendizaje y fase de recuperación.

CAPÍTULO 2. ESTADO DEL ARTE

En este capítulo se habla de la investigación realizada previamente a la elaboración del proyecto. En la sección 2.1 se habla de la historia de la traducción automática. En la 2.2 se describen los principales tipos de traducción automática. La historia, características y funcionamiento del traductor de *Google* (uno de los más utilizados a nivel mundial) están descritas en la sección 2.3. Finalmente, los conceptos de funcionamiento de las Memorias de Traducción, así como algunas alternativas comerciales, son analizados en la sección 2.4.

2.1. TRADUCCIÓN AUTOMÁTICA.

^[1]La idea de la traducción automática puede remontarse al siglo XVII. En 1629, René Descartes propuso un lenguaje universal, con las ideas equivalentes en lenguas diferentes que comparten un mismo símbolo. En la década de los 50, el experimento de Georgetown involucraba una traducción totalmente automática de más de sesenta oraciones del ruso al inglés. El experimento fue todo un éxito y marcó el comienzo de una era con un importante financiamiento para la investigación de tecnologías que permitieran la traducción automática. Los autores afirmaban que en un plazo de tres a cinco años, la traducción automática sería un problema resuelto.

En los 80's tanto los ordenadores como las técnicas de programación eran muy rudimentarias (se programaba mediante el cableado de tableros en lenguaje máquina), por lo que las posibilidades reales de probar los métodos eran mínimas. Pero al final de la misma, de la nada hubo una explosión en cuanto a lo tecnológico lo cual logro la expansión de la traducción automática.

2.2. TIPOS DE TRADUCCIÓN AUTOMÁTICA

^[1]Si disponen de suficiente información, las traducciones automáticas pueden funcionar bastante bien, permitiendo que personas con una lengua materna determinada sean capaces de hacerse una idea de lo que ha escrito otra persona en su idioma. El problema principal reside en obtener la información adecuada para cada uno de los métodos de traducción.

Según su aproximación, los sistemas de traducción automática se pueden clasificar entre dos grandes grupos: los que se basan en reglas lingüísticas por una parte, y los que utilizan *corpus* textuales por otra.

2.2.1. TRADUCCIÓN AUTOMÁTICA BASADA EN REGLAS

^[1]La traducción automática mediante reglas consiste en realizar transformaciones a partir del original, reemplazando las palabras por su equivalente más apropiado. A este tipo de transformaciones del texto original se llama predicción de textos.

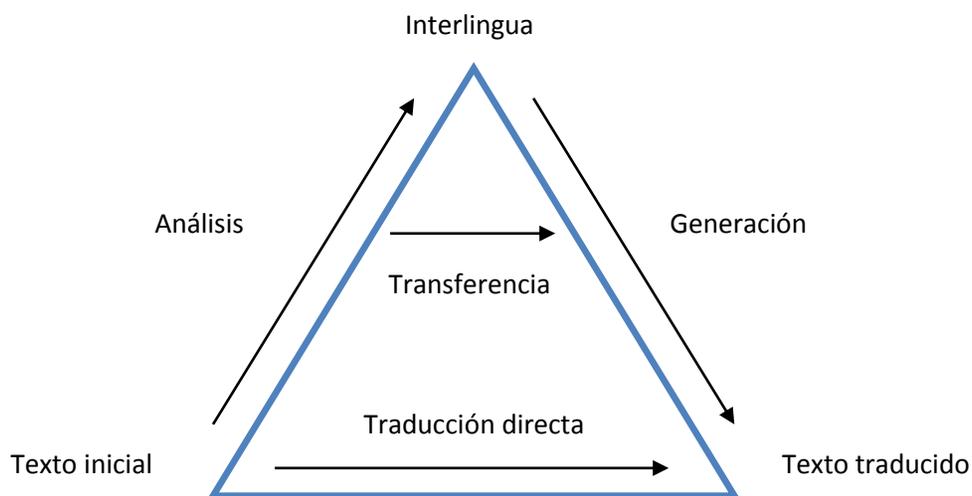


Figura 1: Esquema que muestra la relación entre los diferentes paradigmas de traducción automática basada en reglas.

a) Transferencia

En la traducción por transferencia, el análisis del original juega un papel más importante, y da paso a una representación interna que es la que se utiliza como *enlace* para traducir entre idiomas distintos.

b) Lenguaje intermedio

La traducción automática a partir de un lenguaje intermedio es un caso particular de la traducción automática basada en reglas. El lenguaje original, por ejemplo un texto que debe ser traducido, es transformado a un lenguaje intermedio, cuya estructura es independiente a la del lenguaje original y a la del lenguaje final. El texto en el lenguaje final se obtiene a partir de la representación del texto en el lenguaje intermedio. En general a esta lengua intermedia se le llama "interlingua".

2.2.2. TRADUCCIÓN AUTOMÁTICA BASADA EN CORPUS.

^[1]La traducción automática a partir de un corpus lingüístico se basa en el análisis de muestras reales con sus respectivas traducciones. Entre los mecanismos que utilizan corpus incluyen los métodos estadísticos y los basados en ejemplos.

a) Estadística

El objetivo de la traducción automática estadística es generar traducciones a partir de métodos estadísticos basados en corpus de textos bilingües, como por ejemplo las actas del parlamento europeo, que se encuentran traducidas en todos los idiomas oficiales de la UE. Si la existencia de estos corpus fuese mayor, se podrían conseguir resultados excelentes al traducir textos de ámbitos similares.

El primer programa de traducción automática estadística fue Candide, desarrollado por IBM.

Antes el Traductor de Google usaba SYSTRAN, pero ahora está usando la traducción estadística. Ya hablaremos de él más adelante.

Aunque la exactitud de las traducciones, tanto las estadísticas como las que no, se ha incrementado con los años, la gran cantidad de posibilidades que tiene una palabra de ser traducida de un idioma a otro relega la traducción automática a un método que tan solo permite transmitir la idea esencial.

b) Basada en ejemplos

La traducción automática basada en ejemplos, se caracteriza por el uso de un corpus bilingüe como principal fuente de conocimiento en tiempo real. Es esencialmente una traducción por analogía y puede ser interpretada como una implementación del razonamiento por casos base empleado en el aprendizaje automático, que consiste en la resolución de un problema basándose en la solución de problemas similares. De aquí parten las Memorias Traductoras las cuales mencionaremos en breve.

c) Traducción automática basada en el contexto

La traducción automática basada en el contexto utiliza técnicas basadas en hallar la mejor traducción para una palabra fijándose en el resto de palabras que la rodean; básicamente este método se basa en tratar el texto en unidades de entre 4 y 8 palabras, de manera que se traduce cada una de ellas por su traducción al idioma destino y se eliminan las traducciones que han generado una frase sin sentido.

Luego se mueve la ventana una posición (es decir, a la siguiente palabra), retraduciendo la mayoría de ellas de nuevo y volviendo a filtrar dejando sólo las frases coherentes. Se repite dicho paso para todo el texto. Después se

concatenan los resultados de dichas ventanas de manera que se logre una única traducción del texto.

El filtrado que se realiza donde se decide si una frase tiene sentido utiliza un corpus del lenguaje destino, donde se cuentan el número de apariciones de la frase buscada. Es por tanto un método basado en ideas bastante simple que ofrece unos muy buenos resultados en comparación a otros métodos.

Como ventajas aporta también la facilidad de añadir nuevas lenguas, ya que sólo es necesario un buen diccionario (que puede ser cualquier versión comercial adaptada mediante reglas gramaticales para tener los verbos conjugados y los nombres/adjetivos con sus variaciones en número y género) y un Corpus en el lenguaje destino, que se puede obtener por ejemplo de Internet.

2.3. EL TRADUCTOR MÁS UTILIZADO EN LA ACTUALIDAD: GOOGLE TRANSLATE

^[9] “Google Translate” o “Traductor de Google” es un sistema de traducción automática gratuito proporcionado por Google Inc. basado en datos estadísticos para traducir texto, documentos y páginas web a otras lenguas. Google introdujo su propio software de traducción en 2007, antes Google utilizaba el traductor SYSTRAN, que es utilizado por otros servicios de traducción, tales como Babel Fish, AOL y Yahoo.

2.3.1. CARACTERÍSTICAS Y LIMITACIONES

El Traductor de Google, al igual que otras herramientas de traducción automática, tiene sus limitaciones. Si bien puede ayudar al lector a comprender el contenido general de un texto en lengua extranjera, no siempre ofrece traducciones precisas. Con algunos idiomas se obtienen mejores resultados que con otros. El rendimiento del traductor mejora cuanto menor es la longitud del texto a traducir, este efecto es particularmente evidente en las traducciones de

chino a inglés. El equipo de investigación de Google ha desarrollado un sistema de traducción estadístico propio que actualmente incluye un sistema de traducción por voz para que las personas con algún tipo de limitación tengan acceso a la información en cualquier idioma.

2.3.2. INTEGRACIÓN CON EL NAVEGADOR

Existen barras de herramientas para varios navegadores y una serie de extensiones de Firefox para los servicios de Google, en particular para el traductor, que permiten el acceso directo al servicio de traducción. También hay disponibles varios Google Gadgets. (Versiones en miniatura del sitio web en cualquier tamaño de anuncio) que utilizan el traductor para permitir integrarlo con iGoogle y otras páginas Web. Los Gadgets interactúan con el usuario y utilizan las API (Interfaz de programación de aplicaciones) de Google para permitir realizar traducciones.

En febrero de 2010 traductor Google se integró en el navegador estándar de Google Chrome.

2.3.3. METODOLOGÍA DE TRADUCCIÓN.

Esta es la parte que más nos interesa del traductor, pues no se aplican reglas gramaticales, ya que sus algoritmos se basan en el análisis estadístico más que en el análisis tradicional basado en reglas. El Traductor Google se basa en un enfoque llamado traducción automática estadística, y más específicamente, en la investigación de Franz-Josef Och que ganó el concurso de DARPA en velocidad de traducción automática en 2003. Och dirige actualmente el departamento de traducción automática de Google. Según el discurso de Keynote Och en la Cumbre de traducción de 2005, una base sólida para el desarrollo de un sistema estadístico utilizable en un sistema de traducción automática para un nuevo par de lenguas, consistiría en tener un cuerpo de texto bilingüe (o una

colección paralela) de más de un millón de palabras y dos cuerpos monolingües cada uno de más de mil millones de palabras. Se utilizan modelos estadísticos de estos datos para traducir entre los idiomas. Para adquirir esta enorme cantidad de datos lingüísticos, Google utiliza los documentos de las Naciones Unidas. El mismo documento está disponible normalmente en los seis idiomas oficiales (árabe, chino, inglés, francés, ruso y español), por lo que el traductor Google dispone de 6 cuerpos de idiomas de "traducciones humanas" de 20 mil millones de palabras.

2.4. MEMORIAS DE TRADUCCIÓN (MT)

^[10] El concepto de memoria de traducción existe desde hace varias décadas. La idea se originó en los años 70 del siglo XX y las primeras implementaciones se produjeron en la década de 1980, pero este tipo de herramientas sólo constituyen una entidad comercial significativa desde finales de 1990. Muchas de las herramientas que están disponibles comercialmente hoy en día se desarrollaron por primera vez como herramientas internas utilizadas por las agencias de traducción para apoyar el trabajo de sus clientes, pero como ahora las memorias de traducción se encuentran con más disponibilidad, su popularidad entre otros traductores profesionales ha crecido de manera constante.

Una MT es un tipo de base de datos lingüística que se utiliza para almacenar los textos originales y sus traducciones. Los textos se dividen en segmentos cortos que a menudo se corresponden con oraciones. Una unidad de traducción se compone de un segmento de texto origen y su equivalente traducido; de manera más simple, una MT puede verse como una lista de segmentos de texto origen explícitamente alineados con sus homólogos de texto destino. La estructura resultante se denomina a veces "corpus paralelo" o "bitexto".

Estas unidades de traducción se almacenan en la base de datos de la MT. Algunos programas de MT sofisticados utilizan un tipo de tecnología llamada red neuronal para almacenar información. Una red neuronal permite que la información se recupere más rápidamente que mediante una técnica de búsqueda secuencial. La idea esencial de un sistema de MT es que permite a un traductor reutilizar o "reciclar" segmentos previamente traducidos, lo que a veces

se llama "aprovechamiento". Aunque el lenguaje es dinámico, es muy repetitivo, y se suelen utilizar expresiones similares para comunicar ideas similares en dominios determinados.

El volumen de las traducciones se incrementa cada día y cada vez es más común que se pidan traducciones de textos con párrafos que ya han traducido con anterioridad en otros proyectos. Anteriormente, la mayoría de los traductores no guardaba los archivos de traducciones previas en formato electrónico y los que lo hacían, solían recogerlos de manera no sistemática o de forma que no se podían buscar fácilmente (por ejemplo, en papel). Incluso manteniendo las copias electrónicas, puede resultar difícil y además puede tomar mucho tiempo localizar el segmento necesario.

Por ejemplo, si un traductor mantiene copias de las traducciones anteriores en los archivos del procesador de textos, probablemente tendría que adoptar una búsqueda de varios pasos parecidos a los enumerados a continuación con el fin de poder reutilizar una traducción anterior:

- Localizar en archivo de texto origen adecuado (que a menudo implica descifrar un nombre de archivo críptico).
- Abrir el archivo de texto origen y utilizar la función de búsqueda del procesador de textos para localizar el segmento apropiado.
- Abrir el archivo de destino (la traducción).
- Desplazarse a través del archivo de destino hasta la ubicación aproximada del segmento en cuestión.
- Comenzar a leer para encontrar el equivalente de destino adecuado.
- Copiar y pegar el segmento destino deseado en la nueva traducción.
- Editar el segmento según proceda.

En muchos casos, los traductores que adoptan este enfoque pueden llegar a pasar tanto tiempo abriendo varios archivos, buscando, copiando, pegando y editando, que habría sido más rápido volver a traducir los segmentos a partir de cero.

En los últimos años, los investigadores y desarrolladores han trabajado intensamente para encontrar una solución a este problema y uno de los resultados de sus esfuerzos ha sido los sistemas de Memorias de Traducción, un nuevo tipo de herramienta TAC (Traducción asistida

por computadora) que permite a los traductores reutilizar, y por lo tanto beneficiarse, del gran volumen de trabajo ya traducido.

2.4.1. FUNCIONAMIENTO DE UN SISTEMA DE MT

Esta tecnología funciona mediante la comparación automática de un texto fuente (o texto de origen) con una base de datos de textos que ya han sido traducidos. Cuando un traductor tiene que traducir un nuevo segmento, el sistema consulta la base de datos MT para ver si este nuevo segmento corresponde a un segmento ya traducido.

2.4.2 CREACIÓN DE UNA MT

Un sistema MT es esencialmente un tipo de base de datos. En otras palabras, es una componente de software que permite a un usuario almacenar y recuperar información. Los recientes avances en la tecnología de base de datos y la potencia de los ordenadores han hecho posible que los sistemas de MT puedan almacenar y recuperar rápidamente grandes cantidades de material ya traducido. Sin embargo, como es el caso de cualquier sistema de base de datos, la información debe proporcionarla el usuario. Por lo tanto, cuando un traductor compra un sistema de MT por vez primera, la base de datos está vacía. La MT resulta útil cuando el traductor comienza a almacenar algunos datos (textos de origen y de destino) en el sistema. De hecho, cuanto mayor sea el número de textos almacenados en la base de datos, mayor será la probabilidad de encontrar algún tipo de coincidencia. Sin embargo, el tamaño no debe emplearse en detrimento de la organización. No es necesario, y puede no ser deseable, construir una sola MT. Los traductores pueden encontrar más útil crear memorias de traducción separadas para los distintos campos de estudio existentes (por ejemplo, con el fin de reducir las coincidencias falsas provocadas por homonimia), o tener diferentes memorias de traducción para los distintos clientes (por ejemplo, para dar cabida a las preferencias terminológicas y estilísticas).

Se ha de tener en cuenta que una mayor MT se traducirá en un mayor número de coincidencias sólo si los textos almacenados contienen vocabulario y estructuras similares a los encontrados en el texto de la nueva fuente. Por lo tanto, si bien puede parecer lógico a primera vista construir una sola y amplia MT que abarque todos los campos temáticos o a todos los clientes, esto puede en realidad llegar a ser un atraso, ya que es probable que sólo un porcentaje de los textos en la MT proporcionen buenas coincidencias.

Por otra parte, existe una mayor probabilidad de hallar "ruido" (por ejemplo, coincidencias que no son útiles, que contienen confusiones con homónimos) y el traductor puede perder una cantidad considerable de tiempo en analizar, eliminar o editar estas coincidencias erróneas.

Hay dos formas principales de introducir traducciones en la base de datos MT: a través de traducción interactiva o a través de la alineación post-traducción. Como veremos, la fortaleza de un método constituye la debilidad del otro: la traducción interactiva tiene potencial para producir un MT de alta calidad, pero (en principio) de bajo volumen, mientras que la alineación post-traducción tiene potencial para producir una MT que es mayor en volumen, pero (posiblemente) de menor calidad.

a) Traducción interactiva

Es la forma más sencilla de construir una MT destinada a los traductores, para llevar a cabo el trabajo de traducción en el entorno de MT, añadiendo unidades de traducción a la memoria a medida que avanzan.

Cada vez que el traductor traduce un segmento fuente de texto, la unidad de traducción emparejada (el segmento de texto de origen y su traducción equivalente) se puede almacenar en la base de datos de MT.

Una vez que un segmento se ha traducido y almacenado, se convierte de inmediato en parte de la MT. Esto significa que si ese segmento (o uno similar) aparece de nuevo en el texto —incluso en la siguiente frase— se

sugiere la traducción anterior al traductor de forma automática. El traductor tiene la opción de aceptar dicha traducción o modificarla.

La creación de una MT través de la traducción interactiva generalmente da como resultado una base de datos de mayor calidad; sin embargo, dependiendo del volumen de trabajo que un traductor produce o del número de traductores que contribuyen, el construir una MT de tamaño considerable utilizando este método puede conllevar una cantidad considerable de tiempo.

Vale la pena señalar, sin embargo, que muchos sistemas de MT pueden conectarse en red, lo que significa que varios traductores pueden contribuir a formar una MT, y el volumen de datos que contiene se puede ampliar con mayor rapidez. En una situación en red, también es posible dar diferentes tipos de privilegios a los usuarios existentes con el fin de ejercer alguna forma de control de calidad. Por ejemplo, todos los usuarios pueden tener permiso para consultar la MT, pero al añadir nuevas unidades de traducción se puede limitar a los revisores o traductores consolidados. Las unidades de traducción realizadas por traductores noveles se pueden enviar a un "tanque de almacenamiento" ("holding tank"), donde un revisor pueda verificarlas antes de ser almacenadas en la MT.

Un inconveniente del enfoque interactivo para la construcción de una MT es que los traductores no pueden reutilizar cualquiera de los trabajos de traducción que hicieron antes de adquirir un sistema de MT. Por lo tanto, otra forma de introducir información en una MT es a través de la alineación post-traducción.

b) Alineación post-traducción

La mayoría de los sistemas de MT también contiene una función para alinear traducciones ya existentes, algo que permite a los traductores reutilizar potencialmente las traducciones de sus archivos (siempre que estos textos estén en formato electrónico). La alineación es el proceso de comparar un texto original y su traducción, haciendo coincidir los segmentos correspondientes, y uniéndolos como unidades de traducción en una MT. Algunos sistemas tratan de hacer la alineación "on the fly" (cuando una búsqueda está en curso); sin embargo, la mayoría de los sistemas requieren que la alineación se lleve a cabo antes de la búsqueda. Para este proceso se puede utilizar una herramienta de alineación automática. Existen algunas limitaciones y los resultados dependen de la idoneidad de los textos para la alineación. Para obtener mejores resultados, los textos de origen y de destino deben tener una estructura similar, sino idéntica.

Las técnicas para la alineación automática incluyen la explotación de la correlación.

Otras técnicas de alineación automática consisten en hacer uso de pares de palabras análogas ("cognate") o traducciones conocidas. En tales casos, con una lectura mecánica bilingüe del léxico se pueden utilizar los equivalentes de traducción como palabras de anclaje para ayudar a unir la fuente y los segmentos destino. Por lo tanto, si una palabra dada aparece en el segmento de origen, el programa de alineación espera encontrar en el segmento destino correspondiente su equivalente en lista. Otros tipos de pistas que pueden utilizarse también para orientar el proceso de alineación son los números, las siglas y el formato (negrita, cursiva, subrayado). Si se encuentra alguna de estas características en un segmento de texto fuente, el programa de alineamiento espera encontrarla asimismo en el correspondiente segmento de texto meta.

2.4.3 TRABAJO CON UNA MT EXISTENTE

Una vez que se ha introducido la información en la base de datos de la MT, ya sea mediante la traducción interactiva o por la alineación post-traducción, el traductor tiene una MT disponible para usar cuando lleve a cabo nuevas traducciones. Hay dos formas principales de trabajar con una MT existente: en modo interactivo y en modo no interactivo.

a) El modo interactivo

A medida que el traductor trabaja con el nuevo segmento fuente segmento a segmento, el sistema de MT intenta hacer coincidir los segmentos almacenados en la base de datos con los nuevos segmentos de texto origen. Si se encuentra una coincidencia, el sistema presenta la traducción anterior al traductor, quien tiene la opción de aceptar, modificar o rechazar esta propuesta. A medida que cada nuevo segmento se traduce, la unidad de traducción se agrega inmediatamente a la MT y está disponible para reutilizar la próxima vez que se detecte un segmento idéntico o similar. De esta manera, la MT se va construyendo conforme el traductor trabaja.

b) Modo no interactivo (por lotes)

La mayoría de los sistemas de MT también permiten la traducción no interactiva, a veces denominada pre-traducción, lo que significa que un usuario puede ejecutar un texto original completo a través del sistema y, cada vez que encuentra una coincidencia exacta, se sustituirá automáticamente el nuevo texto del segmento fuente con la traducción que se almacena en la MT. Los segmentos para los cuales no se encuentra ninguna coincidencia más adelante deberán ser traducidos por un traductor, ya sea humano o un sistema de traducción automática. En cualquier caso, todo el texto debe entonces ser post-editado por un traductor humano para asegurar que las sustituciones realizadas por el sistema eran correctas. Si el traductor realiza cambios en las coincidencias que se insertaron de forma

automática, estos cambios se pueden añadir posteriormente a la MT para mantenerla al día.

2.4.4 MEMORIAS DE TRADUCCIÓN COMERCIALES

a) Trados Translator's Workbench

Es líder en el mercado. Permite trabajar con cualquier tipo de documento con extensión .doc. Se integra con Trados *MultiTerm* para gestión terminológica. Incluye filtros adicionales para trabajar en el formato RTF. Dispone de un módulo de alineación de memorias externo al programa. (*WinAlign*).

Su funcionamiento se basa en la inclusión de una serie de macros y plantillas dentro de Microsoft Word, por lo que su dependencia de este producto es total y no opera de forma autónoma.

b) IBM Translation Manager

Es el programa de memorias de traducción más antiguo, diseñado inicialmente por IBM para sus grupos internos de traducción. Todas sus funciones están integradas en el sistema y no dispone de módulos externos al programa.

Combina su sistema de traducción asistida con funciones de memoria de traducción y consulta de diccionarios.

Integra su propio procesador de textos e incluye un módulo de alineación de memorias. Admite diversos formatos sin limitarse a un procesador de textos específico.

Presenta un alto grado de personalización y flexibilidad. Dispone de una potente función de fusión de segmentos.

Actualmente hay un programa derivado, OpenTM2, que es software libre.

c) Start Transit

Permite abrir varios archivos de modo simultáneo. Se vincula a una aplicación de gestión terminológica externa denominada *Star TermStar*.

Los filtros de formatos presentan un alto grado de personalización.

Permite almacenar un gran número de segmentos en menos espacio que los tradicionales sistemas de memorias de traducción.

Contiene un editor integrado de traducción que puede emular la navegación de menús de Microsoft Word.

d) ATRIL Déjà Vu.

Contiene diversos módulos para creación de proyectos, mantenimiento de base de datos y terminología, conversión y alineación de bases de datos y gestión terminológica (*TermWatch*), todos ellos integrados en el producto.

Incluye la función de propagación de texto por todo el documento de forma automática o manual sin incluir la traducción en la memoria si así se desea.

Combina la memoria de traducción y la base de datos terminológica para generar una propuesta de traducción cuando el sistema no encuentra una coincidencia al 100% o parcial.

Tiene una excelente asistencia postventa.

CAPITULO 3. MARCO TEÓRICO

Es este capítulo hablaremos de las Memorias asociativas, qué son y para qué sirven, profundizando en la sección 3.1 los tecnicismos que manejan dichas Memorias. En el 3.2 se mencionan las Memorias Linear Associator, explicando detalladamente las operaciones realizadas en la fase de aprendizaje y la fase de recuperación. Las Memorias Morfológicas, con todos los pasos para su entendimiento se presentan en la sección 3.3. Por último en la sección 3.4 presentamos el concepto teórico de las Memorias Bidireccionales (BAM). Cada sección está complementada con un ejemplo, para mejorar la comprensión.

3.1. MEMORIAS ASOCIATIVAS

^{[2][3]}La memoria asociativa es el almacenamiento y recuperación de información por asociación con varios tipos de información.

Un dispositivo de almacenamiento de información se llama memoria asociativa si permite recuperar información a partir de conocimiento parcial de su contenido, sin saber su localización de almacenamiento. A veces también se le llama memoria de direccionamiento por contenido.

$$X \rightarrow [M] \rightarrow Y$$

El patrón de entrada se presenta por un vector columna denotado por x y el patrón de salida por un vector columna denotado por y .

Cada uno de estos patrones de entrada forma una asociación con el correspondiente patrón de salida. La notación para una asociación es (x, y) ; en general, para un número entero positivo μ específico, la asociación correspondiente será (x^μ, y^μ) .

La memoria asociativa M se representa mediante una matriz cuya componente es ij –ésima es m_{ij} ; la matriz M se genera a partir de un conjunto de asociaciones conocidas de antemano:

este este es el conjunto fundamental de asociaciones o simplemente conjunto fundamental el cual se representa de la siguiente manera.

$$\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$$

La naturaleza del conjunto fundamental proporciona un importante criterio para clasificar las Memorias asociativas. Si se cumple que $y^\mu = x^\mu \forall \mu \in \{1, 2, \dots, p\}$, se dice que la memoria es autoasociativa; de otro modo, la memoria es heteroasociativa (Kohonen, 1972). Para una memoria heteroasociativa se puede afirmar lo siguiente: $\exists \mu \in \{1, 2, \dots, p\}$ para el que se cumple que $y^\mu \neq x^\mu$.

Tipos de memorias asociativas:

- Memorias heteroasociativas: establecen una correspondencia de x (vector de entrada) en y (vector de salida), de distinta dimensión. Dichos patrones se llaman memorias de referencia.
- Memorias autoasociativas: establece la misma correspondencia que la memoria heteroasociativa pero siendo los patrones de entrada y de salida los mismos

Una memoria asociativa consta de 2 etapas, la primera es la fase de aprendizaje y la segunda es la fase de recuperación.

En la fase de aprendizaje es donde se genera la memoria asociativa a partir de las p asociaciones del conjunto fundamental, y la fase de recuperación es donde la memoria asociativa opera sobre un patrón de entrada.

Sean m, n números enteros positivos; se denota n la dimensión de los patrones de entrada, y por m la dimensión de los patrones de salida; nada impide que los valores de m y n sean iguales. Sin embargo uno de los requisitos que debe cumplir una memoria autoasociativa es que la dimensión de los patrones de entrada sea igual a la dimensión de los patrones de salida; por otro lado, si en una memoria sucede que $m \neq n$, es evidente que la memoria debe ser heteroasociativa.

A fin de especificar las componentes de los patrones, se requiere la notación para dos conjuntos a los que llamaremos arbitrariamente A y B. Las componentes de los vectores columna que representan a los patrones, tanto de entrada como de salida, serán elementos del conjunto A, y las entradas de la matriz M serán elementos del conjunto B.

Cada vector columna que representa a un patrón de entrada tiene n componentes cuyos valores pertenecen al conjunto A, y cada vector columna que representa a un patrón de salida posee m componentes cuyos valores pertenecen al conjunto A. Es decir:

$$x^\mu \in A^n \ \& \ y^\mu \in A^m \ \forall \mu \in \{1, 2, \dots, p\}$$

La *j-ésima* componente de un vector columna se indica con la misma letra del vector, pero sin negrilla, colocando a *j* como subíndice ($j \in \{1, 2, \dots, n\}$ o $j \in \{1, 2, \dots, m\}$ según corresponda). La *j-ésima* componente de un vector columna x^μ se representa por:

$$x_j^\mu$$

Al usar el superíndice t para indicar el transpuesto de un vector, se obtienen las siguientes expresiones para los vectores columna que representan a los patrones fundamentales de entrada y de salida, respectivamente:

$$x^\mu = (x_1^\mu, \ x_2^\mu, \ \dots \ x_n^\mu)^t = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n$$

$$y^\mu = (y_1^\mu, \ y_2^\mu, \ \dots \ y_m^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

Las computadoras tradicionales no usan este direccionamiento; se basan en el conocimiento exacto de la dirección de memoria en la que se encuentra la información.

Sin embargo, se cree que el cerebro humano no actúa así. Si queremos recordar el nombre de una persona, no nos sirve saber que fue el nombre número 3274 que aprendimos. Es más útil saber que su nombre empieza y termina por 'N' y que es un famoso científico inglés. Con esta información, es casi seguro que recordaremos exitosamente a "Newton".

Las memorias asociativas son una de las redes neuronales artificiales más importantes con un amplio rango de aplicaciones en áreas tales como: Memorias de acceso por contenido, identificación de patrones y control inteligente.

Con todo lo visto anteriormente, ya es posible comprender los tipos de memorias asociativas que presentaremos a continuación.

3.2. MEMORIAS LINEAR ASSOCIATOR

^[4]Respecto a la creación de este modo de memoria asociativa, es pertinente mencionar un hecho curioso, que se ha presentado en personajes dedicados a otras ramas de la ciencia: James A Anderson y Teuvo Kohonen obtuvieron resultados asombrosamente similares a pesar de que trabajaron independientemente y sin tener noticia uno del otro, hasta tiempo después de que aparecieron los artículos; además, estos autores tienen formaciones profesionales totalmente diferentes: Anderson es neurofisiólogo y Kohonen es físico e ingeniero eléctrico (Anderson & Rosenfeld,1990; Kohonen, 1989).

Para representar el Linear Associator, consideremos el conjunto fundamental $\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ con $A = \{0, 1\}$:

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n$$

$$y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

3.2.1. FASE DE APRENDIZAJE

Para cada una de las p asociaciones (x^μ, y^μ) se encuentra la matriz $y^\mu (x^\mu)^t$ de dimensiones $m \times n$.

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu \quad x_2^\mu \quad \dots \quad x_n^\mu)$$

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \dots & y_1^\mu x_j^\mu & \dots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \dots & y_2^\mu x_j^\mu & \dots & y_2^\mu x_n^\mu \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \dots & y_i^\mu x_j^\mu & \dots & y_i^\mu x_n^\mu \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \dots & y_m^\mu x_j^\mu & \dots & y_m^\mu x_n^\mu \end{pmatrix}$$

Se suman las p matrices para obtener la memoria:

$$M = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n}$$

De manera que la ij -ésima componentes de la matriz M se expresa así:

$$m_{ij} = \sum_{\mu=1}^p y_i^\mu x_j^\mu$$

3.2.2. FASE DE RECUPERACIÓN

La fase de recuperación consiste en presentarle a la memoria un patrón de entrada x^ω donde $\omega \in \{1, 2, \dots, p\}$:

$$M \cdot x^\omega = \left[\sum_{\mu=1}^p y^\mu \cdot (x^\omega)^t \right] \cdot x^\omega$$

Para que se tenga una recuperación perfecta es necesario que los patrones de entrada sean ortonormales, esto es

$$(x^\mu)^t \cdot x^\omega = \delta_{\mu\omega} = \begin{cases} 1 & \text{si } \mu = \omega \\ 0 & \text{si } \mu \neq \omega \end{cases}$$

donde $\delta_{\mu\omega}$ es la conocida delta de Kronecker

En matemática, la **delta de Kronecker** es una función de dos variables, que vale 1 si son iguales, y 0 si son diferentes. Se escribe con el símbolo δ_{ij} .

3.2.3. EJEMPLO DE LINEAR ASSOCIATOR

Se tiene 2 parejas de patrones ($p=2$) donde los patrones de entrada son ortonormales (es ortonormal si es a la vez un conjunto ortogonal y la norma de cada uno de sus vectores es igual a 1) y tienen dimensión $n=2$. Los patrones de salida tiene una dimensión de $m=6$.

$$x^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad y^1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad y^2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

a) Fase de aprendizaje: Asociaciones

$$y^1 \cdot (x^1)^t = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \cdot (1 \ 0)$$

$$y^1 \cdot (x^1)^t = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

$$y^1 \cdot (x^1)^t = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \cdot (0 \ 1)$$

$$y^1 \cdot (x^1)^t = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$M = \sum_{\mu=1}^2 y^{\mu} \cdot (x^{\mu})^t$$

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Memoria Asociativa:

$$MA = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

b) Fase de recuperación

$$M \cdot x^1 = \left[\sum_{\mu=1}^2 y^{\mu} \cdot (x^{\mu})^t \right] \cdot x^1$$

$$M \cdot x^1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = y^1$$

De igual manera se utiliza la misma dinámica para obtener el patrón 2

$$M \cdot x^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = y^2$$

3.3. MEMORIAS ASOCIATIVAS MORFOLÓGICAS

^[5-8]La diferencia fundamental entre las memorias asociativas clásicas (Lernmatrix, Correlograph, Linear Associator y Memoria Asociativa Hopfield) y las memorias asociativas morfológicas radica en los fundamentos operacionales de éstas últimas, que son las operaciones morfológicas de dilatación y erosión; el nombre de las memorias asociativas morfológicas está inspirado precisamente en estas dos operaciones básicas. Estas memorias rompieron con el esquema utilizado a través de los años en los modelos de memorias

asociativas clásicas, que utilizan operaciones convencionales entre vectores y matrices para la fase de aprendizaje y suma de productos para la recuperación de patrones. Las memorias asociativas morfológicas cambian los productos por sumas y las sumas por máximos o mínimos en ambas fases, tanto de aprendizaje como de recuperación.

Tipos de Memorias asociativas Morfológicas.

$$\text{Máxima} \rightarrow M \rightarrow \vee$$

$$\text{Mínima} \rightarrow W \rightarrow \wedge$$

3.3.1. FASE DE APRENDIZAJE

Sea el conjunto fundamental $\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ con $x \in R^n$ y $y \in R^m$ realizando cada asociación de la siguiente manera:

$$y^\mu \otimes (-x^\mu)^t = \begin{bmatrix} y_1^\mu - x_1^\mu & \cdots & y_1^\mu - x_n^\mu \\ \vdots & \ddots & \vdots \\ y_m^\mu - x_1^\mu & \cdots & y_m^\mu - x_n^\mu \end{bmatrix}$$

A los patrones que conforman el conjunto fundamental se les llama patrones fundamentales.

Las memorias Máximas y mínimas se obtienen aplicando la operación morfológica de erosión:

$$M = \bigvee_{\mu=1}^p (y^\mu \otimes (-x^\mu)^t)$$

$$W = \bigwedge_{\mu=1}^p (y^\mu \otimes (-x^\mu)^t)$$

3.3.2. FASE DE RECUPERACIÓN

En este caso se aplican las operaciones de dilatación:

$$M\Delta x = \begin{bmatrix} \bigwedge_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigwedge_{i=1}^n (y_m - x_i + x_i) \end{bmatrix} = y$$

$$W\nabla x = \begin{bmatrix} \bigvee_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigvee_{i=1}^n (y_m - x_i + x_i) \end{bmatrix} = y$$

3.3.3. EJEMPLO DE UNA MEMORIA ASOCIATIVA MORFOLÓGICA

Teniendo las siguientes 2 ($p=2$) parejas de vectores (x, y) con dimensiones de $n=2$ y $m=5$, respectivamente:

$$x^1 = \begin{pmatrix} 6 \\ 2 \end{pmatrix} \rightarrow y^1 = \begin{pmatrix} 3 \\ 5 \\ 2 \\ -1 \\ -2 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 8 \\ 5 \end{pmatrix} \rightarrow y^2 = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 8 \\ 3 \end{pmatrix}$$

a) Fase de Aprendizaje

- $p=1$

$$y^1 \otimes (-x^1)^t = \begin{pmatrix} 3 \\ 5 \\ 2 \\ -1 \\ -2 \end{pmatrix} \otimes (6 \ 2)$$

$$y^1 \otimes (-x^1)^t = \begin{pmatrix} 3-6 & 3-2 \\ 5-6 & 5-2 \\ 2-6 & 2-2 \\ -1-6 & -1-2 \\ -2-6 & -2-2 \end{pmatrix}$$

$$y^1 \otimes (-x^1)^t = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ -7 & -3 \\ -8 & -4 \end{pmatrix}$$

- $p=2$

$$y^2 \otimes (-x^2)^t = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 8 \\ 3 \end{pmatrix} \otimes (8 \ 5)$$

$$y^2 \otimes (-x^2)^t = \begin{pmatrix} 5-8 & 5-5 \\ 1-8 & 1-5 \\ 0-8 & 0-5 \\ 8-8 & 8-5 \\ 3-8 & 3-5 \end{pmatrix}$$

$$y^2 \otimes (-x^2)^t = \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ 0 & 3 \\ -5 & -2 \end{pmatrix}$$

- Memoria MÁX

$$M = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ -7 & -3 \\ -8 & -4 \end{pmatrix} \vee \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ 0 & 3 \\ -5 & -2 \end{pmatrix}$$

$$M = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ 0 & 3 \\ -5 & -2 \end{pmatrix}$$

- Memoria MÍN

$$M = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ -7 & -3 \\ -8 & -4 \end{pmatrix} \wedge \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ 0 & 3 \\ -5 & -2 \end{pmatrix}$$

$$M = \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ -7 & -3 \\ -8 & -4 \end{pmatrix}$$

b) Fase de Recuperación

- Presentando el primer patrón la memoria MÁX

$$M\Delta x^1 = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ 0 & 3 \\ -5 & -2 \end{pmatrix} \Delta \begin{pmatrix} 6 \\ 2 \end{pmatrix}$$

$$M\Delta x^1 = \begin{pmatrix} (-3+6) & \wedge & (1+2) \\ (-1+6) & \wedge & (3+2) \\ (-4+6) & \wedge & (0+2) \\ (0+6) & \wedge & (3+2) \\ (-5+6) & \wedge & (-2+2) \end{pmatrix}$$

$$M\Delta x^1 = \begin{pmatrix} 3 & \wedge & 3 \\ 5 & \wedge & 5 \\ 2 & \wedge & 2 \\ 6 & \wedge & 5 \\ 1 & \wedge & 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \\ 2 \\ 5 \\ 0 \end{pmatrix} \neq y^1$$

- Presentando a la Memoria MÍN

$$W\Delta x^1 = \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ -7 & -3 \\ -8 & -4 \end{pmatrix} \Delta \begin{pmatrix} 6 \\ 2 \end{pmatrix}$$

$$W\Delta x^1 = \begin{pmatrix} (-3+6) & \vee & (0+2) \\ (-7+6) & \vee & (-4+2) \\ (-8+6) & \vee & (-5+2) \\ (-7+6) & \vee & (-3+2) \\ (-8+6) & \vee & (-4+2) \end{pmatrix}$$

$$W\Delta x^1 = \begin{pmatrix} 3 & \vee & 2 \\ -1 & \vee & -2 \\ -2 & \vee & -3 \\ -1 & \vee & -1 \\ -2 & \vee & -2 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ -2 \\ -1 \\ -2 \end{pmatrix} \neq y^1$$

- Presentando el patrón numero dos a la memoria MÁX

$$M\Delta x^2 = \begin{pmatrix} -3 & 1 \\ -1 & 3 \\ -4 & 0 \\ 0 & 3 \\ -5 & -2 \end{pmatrix} \Delta \begin{pmatrix} 8 \\ 5 \end{pmatrix}$$

$$M\Delta x^2 = \begin{pmatrix} (-3+8) \wedge (1+5) \\ (-1+8) \wedge (3+5) \\ (-4+8) \wedge (0+5) \\ (0+8) \wedge (3+5) \\ (-5+8) \wedge (-2+5) \end{pmatrix}$$

$$M\Delta x^2 = \begin{pmatrix} 5 \wedge 6 \\ 7 \wedge 8 \\ 4 \wedge 5 \\ 8 \wedge 8 \\ 3 \wedge 3 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 4 \\ 8 \\ 3 \end{pmatrix} \neq y^2$$

- Presentando a la Memoria MÍN

$$W\Delta x^2 = \begin{pmatrix} -3 & 0 \\ -7 & -4 \\ -8 & -5 \\ -7 & -3 \\ -8 & -4 \end{pmatrix} \Delta \begin{pmatrix} 8 \\ 5 \end{pmatrix}$$

$$W\Delta x^2 = \begin{pmatrix} (-3+8) \vee (0+5) \\ (-7+8) \vee (-4+5) \\ (-8+8) \vee (-5+5) \\ (-7+8) \vee (-3+5) \\ (-8+8) \vee (-4+5) \end{pmatrix}$$

$$W\Delta x^2 = \begin{pmatrix} 5 \vee 5 \\ 1 \vee 1 \\ 0 \vee 0 \\ 1 \vee 2 \\ 0 \vee 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix} \neq y^2$$

3.4. VECTOR ONE HOT

Arreglo de vectores Ortonormales que están compuestos por n-bits donde solo uno de ellos tiene valor alto ejemplo:

$$\text{Vector One Hot} = (0 \ 1 \ 0 \ 0 \ 0 \ 0)$$

Esta técnica de One Hot es usada para etiquetar una matriz de $n \times m$ valores de la siguiente manera:

$$\text{Matriz original} = \begin{pmatrix} 1 & 5 & 6 \\ 2 & 3 & 8 \\ 3 & 0 & 4 \end{pmatrix}$$

$$\text{Matriz One Hot} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Matriz Resultante} = \begin{pmatrix} 1 & 5 & 6 \\ 2 & 3 & 8 \\ 3 & 0 & 4 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3.5. VECTORES ZERO HOT

Arreglo de vectores que están compuestos por n -bits donde solo uno de ellos tiene valor bajo ejemplo:

$$\text{Vector Zero Hot} = (1 \ 0 \ 1 \ 1 \ 1 \ 1)$$

Esta técnica de Zero Hot es usada de igual manera para etiquetar una matriz de $n \times m$ valores:

$$\text{Matriz original} = \begin{pmatrix} 3 & 4 & 2 \\ 6 & 0 & 7 \\ 7 & 9 & 6 \end{pmatrix}$$

$$\text{Matriz Zero Hot} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$\text{Matriz Resultante} = \begin{pmatrix} 3 & 4 & 2 \\ 6 & 0 & 7 \\ 7 & 9 & 6 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

3.6. MEMORIAS BAM

En general, cualquier modelo de memoria asociativa bidireccional se podría esquematizar como se muestra en la figura 3.1



Figura 2: Esquema general de una Memoria Asociativa Bidireccional.

La BAM general es una “caja negra” que opera de la siguiente forma: dado un patrón “**x**” obtiene el patrón asociado “**y**”, y dado el patrón “**y**” obtiene el patrón asociado “**x**”. Además, si se asume que \bar{x} y \bar{y} son las versiones ruidosas de “**x**” y “**y**”, respectivamente, se espera que la BAM recupere los patrones correspondientes, “**x**” y “**y**”, libres de ruido.

CAPÍTULO 4. CONSTRUCCIÓN DEL PROYECTO

En este capítulo se describe el nuevo modelo de memoria asociativa bidireccional que se propone como tema central de este trabajo: las Memorias Asociativas Bidireccionales Morfológicas aplicadas a un traductor bilingüe.

4.1. DESCRIPCIÓN DE LAS MEMORIAS ASOCIATIVAS BIDIRECCIONALES MORFOLÓGICAS

En esta sección se describen de manera general el funcionamiento del nuevo modelo, este modelo va relacionado con el algoritmo de la sección.

4.1.1. FASE DE APRENDIZAJE

a) Sentido $x \rightarrow y$

Para comenzar concatenamos cada uno de los vectores de entrada con los vectores One Hot y Zero Hot. De esta manera se obtendrán dos conjuntos de vectores a los que llamaremos vectores "X" los formados con One Hot y \bar{X} los formados con Zero Hot.

$$x^\mu = [x_1^\mu, x_2^\mu, \dots, x_{n-1}^\mu, x_n^\mu | bajo_1^\mu, bajo_2^\mu, \dots, bajo_{m-1}^\mu, alto_m^\mu]$$

Donde el valor alto existe cuando se cumple la condición de $m = \mu$. Ejemplificando tenemos que:

$$x^1 = [x_1^1, x_2^1, \dots, x_{n-1}^1, x_n^1 | alto_1^1, bajo_2^1, \dots, bajo_{m-1}^1, bajo_m^1]$$

$$x^5 = [x_1^5, x_2^5, \dots, x_5^5, \dots, x_{n-1}^5, x_n^5 | bajo_1^5, bajo_2^5, \dots, alto_5^5, \dots, bajo_{m-1}^5, bajo_m^5]$$

De una vez realizado esto con todos los vectores “X” se hará de igual manera con los vectores “ \bar{X} ”. Estos se concatenan con los vectores Zero Hot:

$$\bar{x}^\mu = [\bar{x}_1^\mu, \bar{x}_2^\mu, \dots, \bar{x}_{n-1}^\mu, \bar{x}_n^\mu | alto_1^\mu, alto_2^\mu, \dots, alto_{m-1}^\mu, bajo_m^\mu]$$

Donde el valor bajo tomara posición cuando se cumpla la condición de $m = \mu$

En esta se ocupa la fórmula vista en la sección 3.3 del capítulo anterior. En la fórmula hay un ligero cambio ya que no se ocupa el patrón “Y” para realizar la Matriz sino se auto asocian los valores “X” y “ \bar{X} ”:

$$x^\mu \otimes (-x^\mu)^t = \begin{bmatrix} x_1^\mu - x_1^\mu & \dots & x_1^\mu - x_n^\mu \\ \vdots & \ddots & \vdots \\ x_m^\mu - x_1^\mu & \dots & x_m^\mu - x_n^\mu \end{bmatrix}$$

$$\bar{x}^\mu \otimes (-\bar{x}^\mu)^t = \begin{bmatrix} \bar{x}_1^\mu - \bar{x}_1^\mu & \dots & \bar{x}_1^\mu - \bar{x}_n^\mu \\ \vdots & \ddots & \vdots \\ \bar{x}_m^\mu - \bar{x}_1^\mu & \dots & \bar{x}_m^\mu - \bar{x}_n^\mu \end{bmatrix}$$

Obteniendo todos los patrones de entrada modificados pasaremos a formar nuestras Memorias asociativas MÁX y MÍN aplicando la operación morfológica de erosión, que se menciona en la sección 3.3

$$V = M = \bigvee_{\mu=1}^p (x^\mu \otimes (-x^\mu)^t)$$

$$\Lambda = W = \bigwedge_{\mu=1}^p (\bar{x}^\mu \otimes (-\bar{x}^\mu)^t)$$

Una vez obtenidas las Memorias Máximas y Mínimas con los vectores de entrada entonces finalizara esta etapa con los patrones de salida “Y” que se verán modificados con el método Linear Associator visto en la sección 3.2.

Creamos una matriz cuadrada A ($n \times n$) la cual tendrá los valores de la matriz identidad; su dimensión será el número de patrones de salida (n).

Quedando de manera resumida la matriz de asociación:

$$LMAy = \begin{pmatrix} y_1^1 & y_1^2 & \dots & y_1^{n-1} & \dots & y_1^n \\ y_2^1 & y_2^2 & \dots & y_2^{n-1} & \dots & y_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_i^1 & y_i^2 & \dots & y_i^{n-1} & \dots & y_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_m^1 & y_m^2 & \dots & y_m^{n-1} & \dots & y_m^n \end{pmatrix}$$

b) Sentido $y \rightarrow x$

De igual manera que en sentido $x \rightarrow y$ concatenamos cada uno de los vectores de entrada con los vectores One Hot y Zero Hot. A diferencia que los dos conjuntos de vectores obtenidos los llamaremos vectores “Y” y \bar{Y} .

$$y^\mu = [y_1^\mu, y_2^\mu, \dots, y_{n-1}^\mu, y_n^\mu | bajo_1^\mu, bajo_2^\mu, \dots, bajo_{m-1}^\mu, alto_m^\mu]$$

Donde el valor alto existe cuando se cumple la condición de $m = \mu$. Ejemplificando tenemos que:

$$y^1 = [y_1^1, y_2^1, \dots, y_{n-1}^1, y_n^1 | alto_1^1, bajo_2^1, \dots, bajo_{m-1}^1, bajo_m^1]$$

$$y^6 = [y_1^6, y_2^6, \dots, y_6^6, \dots, y_{n-1}^6, y_n^6 | bajo_1^6, bajo_2^6, \dots, alto_6^6, \dots, bajo_{m-1}^6, bajo_m^6]$$

De una vez realizado esto con todos los vectores “Y” se hará de igual manera con los vectores “ \bar{Y} ”. Estos se concatenan con los vectores Zero Hot

$$\bar{y}^\mu = [\bar{y}_1^\mu, \bar{y}_2^\mu, \dots, \bar{y}_{n-1}^\mu, \bar{y}_n^\mu | alto_1^\mu, alto_2^\mu, \dots, alto_{m-1}^\mu, bajo_m^\mu]$$

Donde el valor bajo tomara posición cuando se cumpla la condición de $m = \mu$
Continuamos autoasociación de los valores “Y” y “ \bar{Y} ”:

$$y^\mu \otimes (-y^\mu)^t = \begin{bmatrix} y_1^\mu - y_1^\mu & \cdots & y_1^\mu - y_n^\mu \\ \vdots & \ddots & \vdots \\ y_m^\mu - y_1^\mu & \cdots & y_m^\mu - y_n^\mu \end{bmatrix}$$

$$\bar{y}^\mu \otimes (-\bar{y}^\mu)^t = \begin{bmatrix} \bar{y}_1^\mu - \bar{y}_1^\mu & \cdots & \bar{y}_1^\mu - \bar{y}_n^\mu \\ \vdots & \ddots & \vdots \\ \bar{y}_m^\mu - \bar{y}_1^\mu & \cdots & \bar{y}_m^\mu - \bar{y}_n^\mu \end{bmatrix}$$

Obteniendo todos los patrones de entrada modificados pasaremos a formar nuestras Memorias asociativas MÁX y MÍN aplicando la operación morfológica de erosión, que se menciona en la sección 3.3

$$V = M = \bigvee_{\mu=1}^p (y^\mu \otimes (-y^\mu)^t)$$

$$\Lambda = W = \bigwedge_{\mu=1}^p (\bar{y}^\mu \otimes (-\bar{y}^\mu)^t)$$

Una vez obtenidas las Memorias Máximas y Mínimas se hace la misma operación que en la sección anterior solo que ocupando el vector "X" como salida. Creando otra matriz cuadrada A ($n \times n$) la cual tendrá la forma de la matriz identidad; su dimensión será el número de patrones de salida (n).

Quedando de manera resumida la matriz de asociación:

$$LMAx = \begin{pmatrix} x_1^1 & x_1^2 & \cdots & x_1^{n-1} & \cdots & x_1^n \\ x_2^1 & x_2^2 & \cdots & x_2^{n-1} & \cdots & x_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^{n-1} & \cdots & x_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_m^1 & x_m^2 & \cdots & x_m^{n-1} & \cdots & x_m^n \end{pmatrix}$$

4.1.2. FASE DE RECUPERACIÓN

a) Sentido $x \rightarrow y$

Para la recuperación de “Y” patrón se espera recibir un patrón de entrada “X” con el cual se crearan dos vectores ruidosos que serán representados con las variables A y B dichos vectores serán creados con el patrón “X” y la concatenación de los vectores U y \bar{U} .

Donde U va a ser igual a:

$$U = \begin{pmatrix} alto_1 \\ alto_2 \\ \vdots \\ alto_n \end{pmatrix}$$

$$\bar{U} = \begin{pmatrix} bajo_1 \\ bajo_2 \\ \vdots \\ bajo_n \end{pmatrix}$$

Quedando expandidos de la siguiente manera:

$$A = \begin{pmatrix} x_1^\mu \\ \vdots \\ x_m^\mu \\ alto_1 \\ \vdots \\ alto_n \end{pmatrix}$$

$$B = \begin{pmatrix} x_1^\mu \\ \vdots \\ x_m^\mu \\ bajo_1 \\ \vdots \\ bajo_n \end{pmatrix}$$

Ahora presentamos A a V y B a Λ para obtener dos nuevos vectores, nos enfocaremos en analizar las contracciones de dichos vectores a los que llamaremos vectores C y D que se podría decir que son los vectores U y \bar{U} alterados por la operación de dilatación.

$$M\Delta x = \begin{bmatrix} \bigwedge_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigwedge_{i=1}^n (y_m - x_i + x_i) \end{bmatrix} = y$$

$$W\nabla x = \begin{bmatrix} \bigvee_{i=1}^n (y_1 - \bar{x}_i + x_i) \\ \vdots \\ \bigvee_{i=1}^n (y_m - \bar{x}_i + x_i) \end{bmatrix} = y$$

Aquí se pueden presentar 3 casos:

- **Caso 1:**

Si el vector $C = One\ Hot$; entonces aplicamos el último paso de Linear Associator.

$$LMAy \cdot C^t = \begin{pmatrix} y_1^1 & y_1^2 & \dots & y_1^{n-1} & \dots & y_1^n \\ y_2^1 & y_2^2 & \dots & y_2^{n-1} & \dots & y_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_i^1 & y_i^2 & \dots & y_i^{n-1} & \dots & y_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_m^1 & y_m^2 & \dots & y_m^{n-1} & \dots & y_m^n \end{pmatrix} \cdot C^t = y$$

- **Caso 2:**

Si $C \neq One\ Hot$, entonces analizamos D , si $D = Zero\ Hot$ aplicamos el último paso de Linear Associator.

$$LMAy \cdot \overline{D^t} = \begin{pmatrix} y_1^1 & y_1^2 & \dots & y_1^{n-1} & \dots & y_1^n \\ y_2^1 & y_2^2 & \dots & y_2^{n-1} & \dots & y_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_i^1 & y_i^2 & \dots & y_i^{n-1} & \dots & y_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_m^1 & y_m^2 & \dots & y_m^{n-1} & \dots & y_m^n \end{pmatrix} \cdot \overline{D^t} = y$$

- **Caso 3:**

Si el vector $C \neq \text{One Hot}$ y el vector $D \neq \text{Zero Hot}$ entonces aplicamos una operación lógica AND.

$$C \text{ AND } \overline{D} = E$$

Después de esto aplicamos el último paso de Linear Associator.

$$LMAy \cdot E^t = \begin{pmatrix} y_1^1 & y_1^2 & \dots & y_1^{n-1} & \dots & y_1^n \\ y_2^1 & y_2^2 & \dots & y_2^{n-1} & \dots & y_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_i^1 & y_i^2 & \dots & y_i^{n-1} & \dots & y_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ y_m^1 & y_m^2 & \dots & y_m^{n-1} & \dots & y_m^n \end{pmatrix} \cdot E^t = y$$

b) Sentido $y \rightarrow x$

Para la recuperación de "X" patrón se espera recibir un patrón de entrada "Y" con el cual se crearan dos vectores ruidosos que serán representados con las variables A y B dichos vectores serán creados con el patrón "Y" y la concatenación de los vectores U y \overline{U} .

Donde U va a ser igual a:

$$U = \begin{pmatrix} alto_1 \\ alto_2 \\ \vdots \\ alto_n \end{pmatrix}$$

$$\bar{U} = \begin{pmatrix} bajo_1 \\ bajo_2 \\ \vdots \\ bajo_n \end{pmatrix}$$

Quedando expandidos de la siguiente manera:

$$A = \begin{pmatrix} y_1^\mu \\ \vdots \\ y_m^\mu \\ alto_1 \\ \vdots \\ alto_n \end{pmatrix}$$

$$B = \begin{pmatrix} y_1^\mu \\ \vdots \\ y_m^\mu \\ bajo_1 \\ \vdots \\ bajo_n \end{pmatrix}$$

Ahora presentamos A a V y B a Λ para obtener dos nuevos vectores, nos enfocaremos en analizar las contracciones de dichos vectores a los que llamaremos vectores C y D que se podría decir que son los vectores U y \bar{U} alterados por la operación de dilatación.

$$M\Delta A = \begin{bmatrix} \bigwedge_{i=1}^n (x_1 - y_i + y_i) \\ \vdots \\ \bigwedge_{i=1}^n (x_m - y_i + y_i) \end{bmatrix} = x$$

$$W\nabla B = \begin{bmatrix} \bigvee_{i=1}^n (x_1 - \bar{y}_i + y_i) \\ \vdots \\ \bigvee_{i=1}^n (x_m - \bar{y}_i + y_i) \end{bmatrix} = x$$

Aquí se pueden presentar 3 casos:

- **Caso 1:**

Si el vector $C = One\ Hot$; entonces aplicamos el último paso de Linear Associator.

$$LMAx \cdot C^t = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^{n-1} & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^{n-1} & \dots & x_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_i^1 & x_i^2 & \dots & x_i^{n-1} & \dots & x_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^{n-1} & \dots & x_m^n \end{pmatrix} \cdot C^t = x$$

- **Caso 2:**

Si $C \neq One\ Hot$, entonces analizamos D, si $D = Zero\ Hot$ aplicamos el último paso de Linear Associator.

$$LMAx \cdot \overline{D^t} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^{n-1} & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^{n-1} & \dots & x_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_i^1 & x_i^2 & \dots & x_i^{n-1} & \dots & x_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^{n-1} & \dots & x_m^n \end{pmatrix} \cdot \overline{D^t} = x$$

- **Caso 3:**

Si el vector $C \neq One\ Hot$ y el vector $D \neq Zero\ Hot$ entonces aplicamos una operación lógica AND.

$$C \text{ AND } \overline{D} = E$$

Después de esto aplicamos el último paso de Linear Associator.

$$LMAx \cdot E^t = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^{n-1} & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^{n-1} & \dots & x_2^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_i^1 & x_i^2 & \dots & x_i^{n-1} & \dots & x_i^n \\ \vdots & \vdots & \cdot & \vdots & \cdot & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^{n-1} & \dots & x_m^n \end{pmatrix} \cdot E^t = x$$

4.2. DESARROLLO DEL PROTOTIPO

En esta sección se describe cómo funciona el sistema del diccionario bilingüe implementado en Matlab. Se describe con diagramas cada una de sus fases de funcionamiento.

En los siguientes diagramas se debe tener en cuenta:

1. Las palabras subrayadas y remarcadas representan los nombres de variables usadas para cálculos y almacenamiento de resultados del algoritmo.
2. Matlab trata a todas las variables que se declaran como matrices que pueden almacenar diferentes tipos de datos: char, int, doublé, etc. (existen otros tipos de datos muy semejantes a los utilizados en el lenguaje C++). Por ejemplo, al guardar la palabra "HOLA" en una variable X, Matlab la transforma en una matriz string de 1 fila y 4 columnas.
3. Para cualquier matriz creada en Matlab, su primer elemento siempre será el (1,1).
4. Matlab ajusta automáticamente el tipo de dato de una matriz, por ejemplo al multiplicar la matriz X (que contiene la palabra "HOLA") por 1, esta se transformara de string a doublé, transformando cada letra en su equivalente ASCII.
5. Matlab también puede cambiar el tamaño de una matriz automáticamente según se necesite.
6. Los nombres de las variables son locales (solo son funcionales en el diagrama actual), por lo que al comparar el diagrama de una función con el de una interna se podrá

percibir que los nombres de las variables de entrada y de salida no siempre coinciden en ambos diagramas. sin embargo el orden de las variables en la llamada de alguna función, significara que en la definición de la misma se recibirán los valores que contengan esas variables en el mismo orden. No se utilizan variables globales; para pasar datos de una función a otra solo se podrá hacer a través de los parámetros de entrada y de salida de las funciones.

7. La simbología de los diagramas es la siguiente:

- Óvalos: Inicio y fin de un algoritmo completo o función.
- Rectángulos: procesos como cálculos o creación de variables.
- Romboide: ingreso e impresión de datos, obtenidos por parte del usuario o por el programa mismo.
- Rombos: decisiones, capaces de representar ciclos *for* o sentencias *if else*.
- Círculos vacíos: conexión de procesos.
- Círculos con incisos: indican que el diagrama continúa en la siguiente página.

4.2.1 DIAGRAMA GENERAL DEL SISTEMA: DICCIONARIO BILINGÜE

En este diagrama se representan todos los procesos que se dan en la función principal del sistema. Es decir, al correr el programa de la interfaz de usuario.

Las variables **Idioma1** e **Idioma2** de tipo *string* guardan los nombres de las bases de datos dependiendo del sentido en el que funcionara el diccionario. Por ejemplo, si el diccionario trabajara como traductor Español-Inglés, la variable **Idioma1** guardara “Español.txt” mientras que **Idioma2** guardara “Inglés.txt”.

El cuadro rojo representa la llamada a la función de aprendizaje del diccionario, con los parámetros de entrada y de salida indicados (al igual que el rectángulo verde que indica la llamada a la función de fase de recuperación del diccionario). Estas funciones se explicaran más adelante.

Figura 3 Diagrama general que muestra el funcionamiento de la interfaz de usuario.

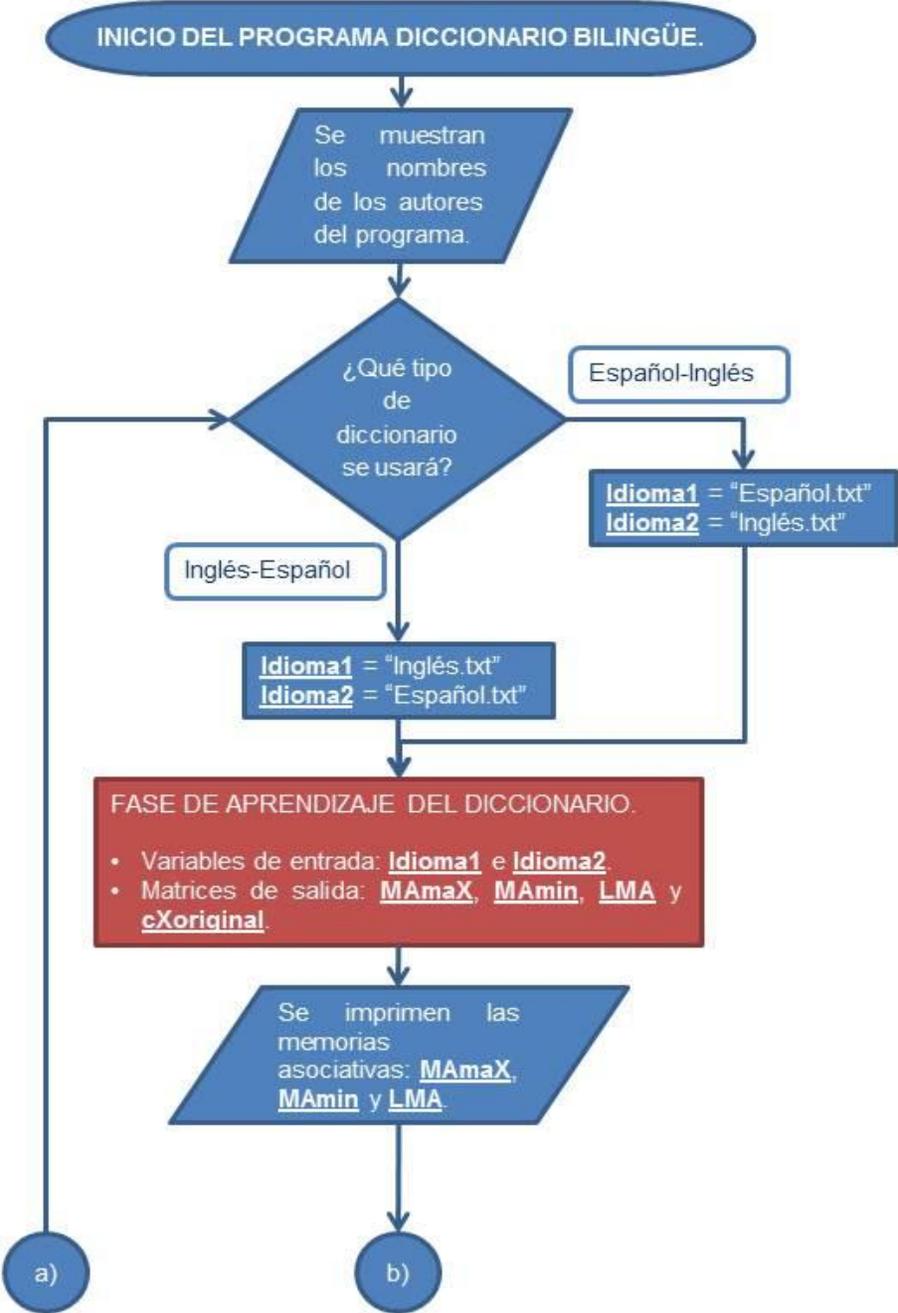


Figura 4: Segunda parte del diagrama anterior.

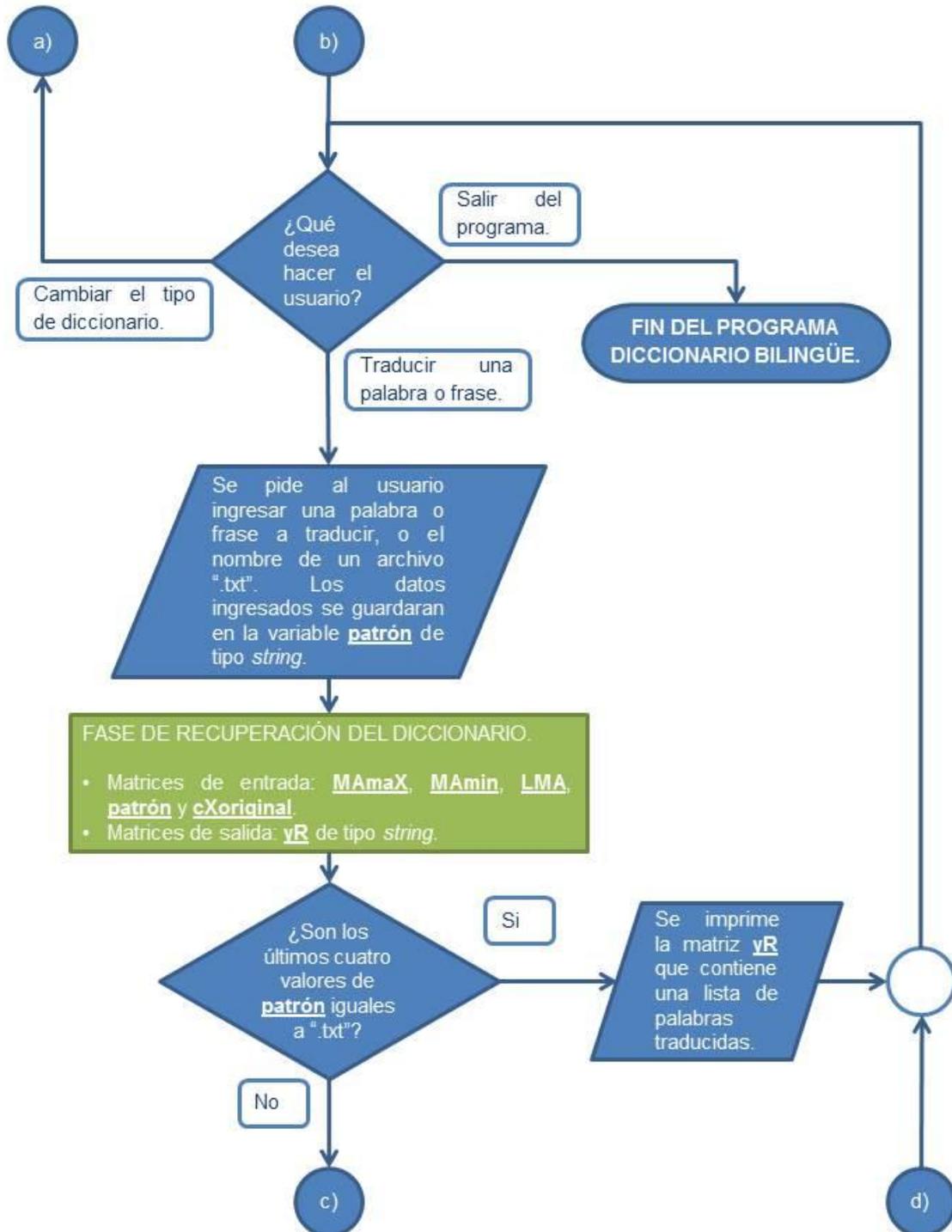
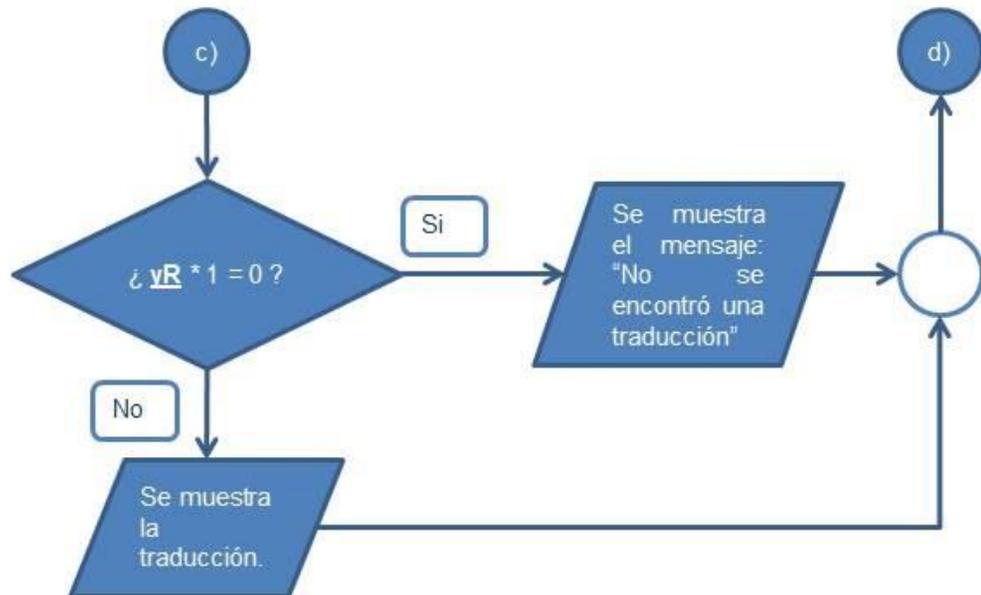


Figura 5: Tercera parte del diagrama.



4.2.2 FASES DE APRENDIZAJE

Para ordenar mejor el código del programa, cada algoritmo de memorias asociativas utilizado se implementó en funciones separadas. También se separó la fase de aprendizaje y recuperación de cada algoritmo. A continuación se describirán todas las fases de aprendizaje usadas en el sistema.

- **FASE DE APRENDIZAJE DEL DICCIONARIO**

El objetivo de esta función es servir de intermediaria entre la interfaz de usuario (diagrama general del sistema) y la fase de aprendizaje principal: la de BAM morfológica.

Figura 6: Primera parte del diagrama de la fase de aprendizaje del diccionario.

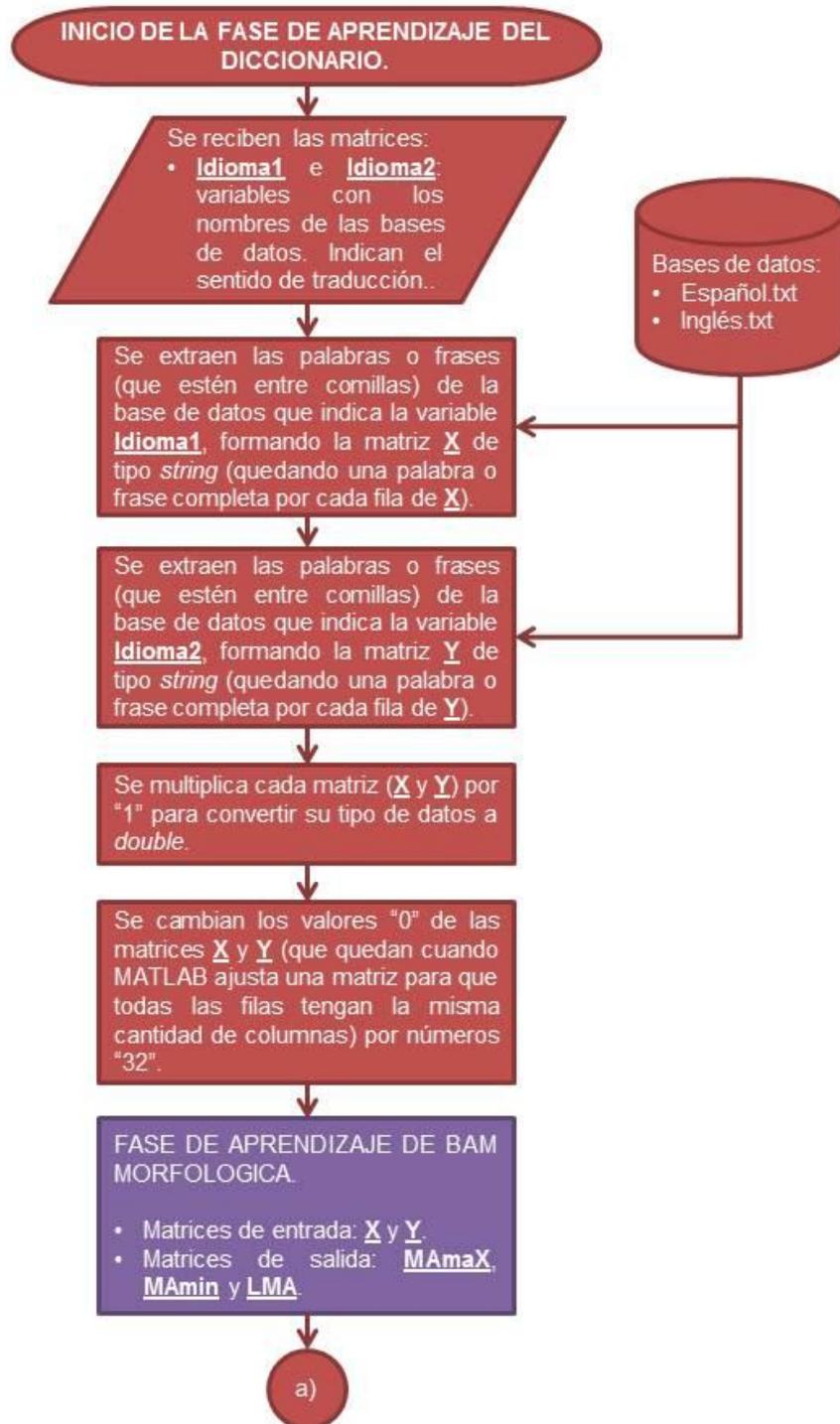


Figura 7: Segunda parte del diagrama de la fase de aprendizaje del diccionario.



- **FASE DE APRENDIZAJE DE BAM MORFOLÓGICA**

Es la función de aprendizaje más importante. Genera las memorias asociativas MAmáx, MAmín y LMA necesarias para las fases de recuperación.

En la parte en la que se forman las matrices Onehot y Zerohot se ve que en lugar de llenar sus diagonales principales con “1” y “0” respectivamente, se usan valores “300” y “-300”. Estos valores representan el rango del valor de umbral elegido, ya que “300” abarca el rango de valores ASCII (de 0 a 255), con esto se logra generar la memoria MAmáx; “-300” es su contraparte negativa necesaria para generar la memoria MAmín con valores negativos. En teoría, cualquier valor de umbral que supere el máximo valor que puedan tener los patrones asociados, será válido.

El cuadro anaranjado indica la llamada a la función de aprendizaje morfológica, aunque algunas de las matrices que esta regresa como resultados no se toman en cuenta para el resto de la función BAM morfológica. El cuadro azul indica la llamada a la función de aprendizaje lineal.

Figura 8: Primera parte del diagrama de la fase de aprendizaje de BAM morfológica.

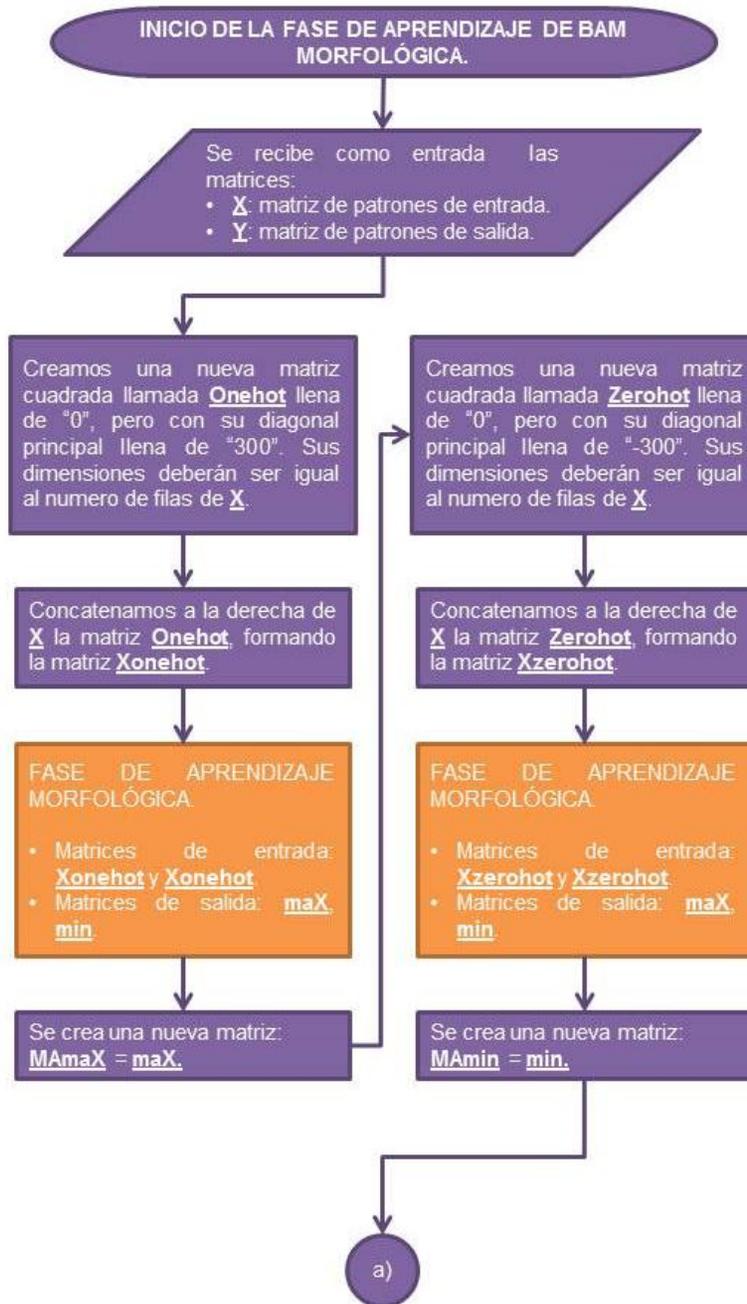
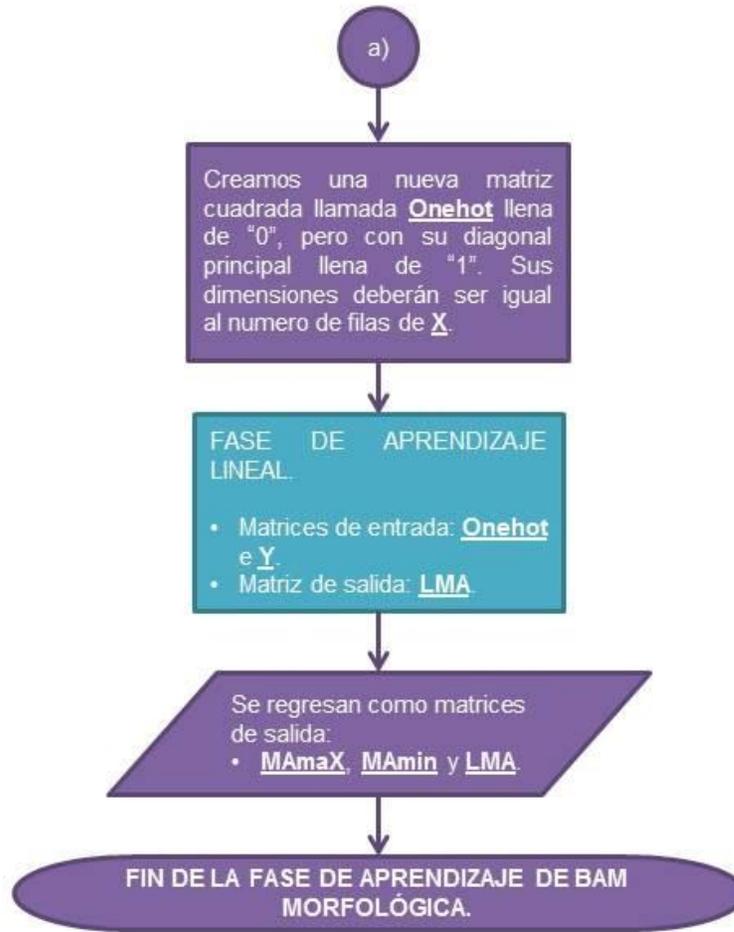


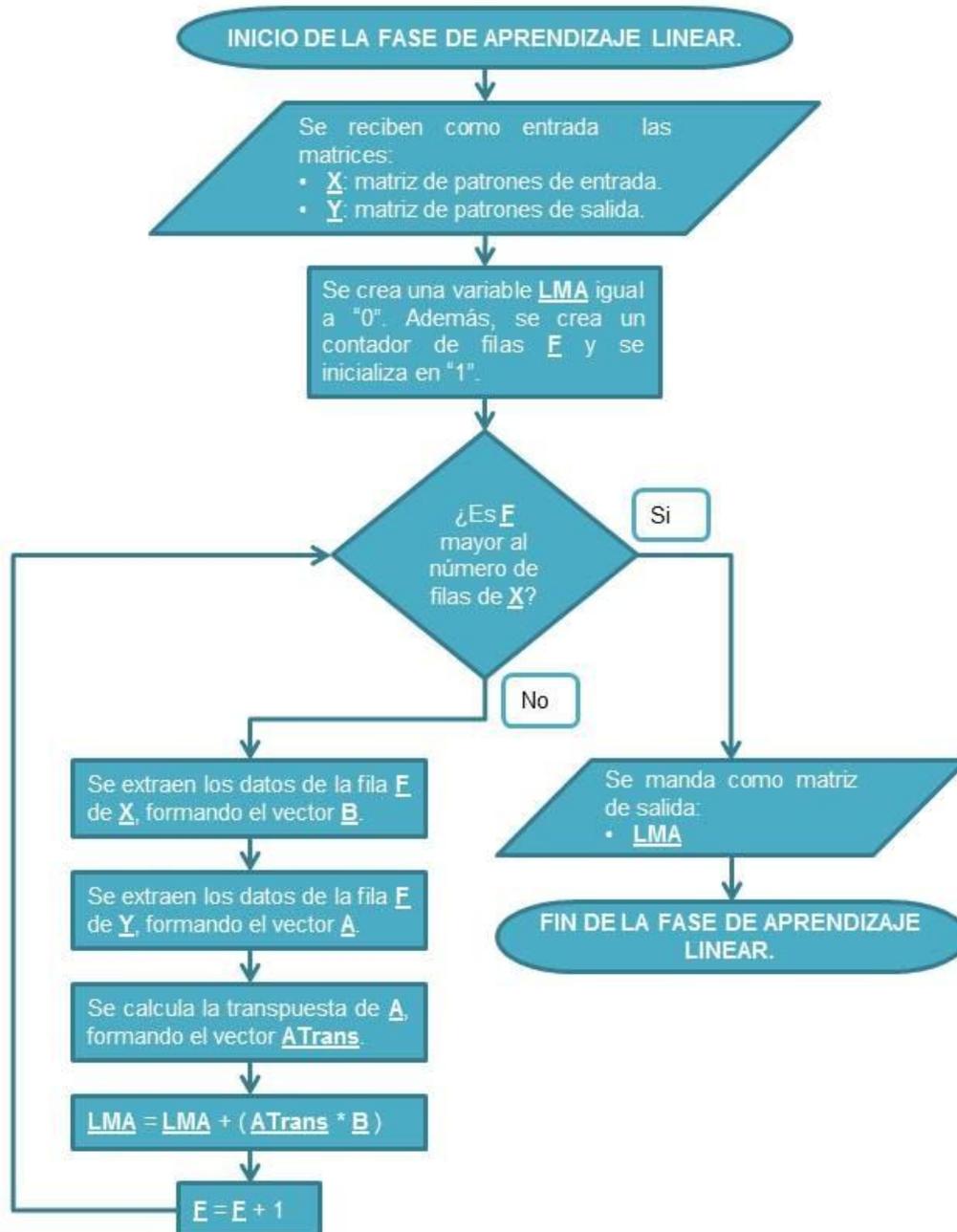
Figura 9: Segunda parte del diagrama de la fase de aprendizaje de BAM morfológica.



- **FASE DE APRENDIZAJE LINEAR**

Esta función genera la memoria asociativa LMA (como detalle, la fase de recuperación lineal nunca se usa en el sistema).

Figura 10: Diagrama de la fase de aprendizaje Linear.



• FASE DE APRENDIZAJE MORFOLÓGICA

Usada por la fase de aprendizaje de BAM morfológica, esta función hace los cálculos para formar las matrices **MAmáx** y **MAmín** a partir de dos matrices de patrones que recibe como parámetros de entrada.

Para funcionar requiere del uso de operaciones de erosión, separadas en las funciones de operador resta, operador máximo y operador mínimo.

Figura 11: Primera parte del diagrama de la fase de aprendizaje morfológica.

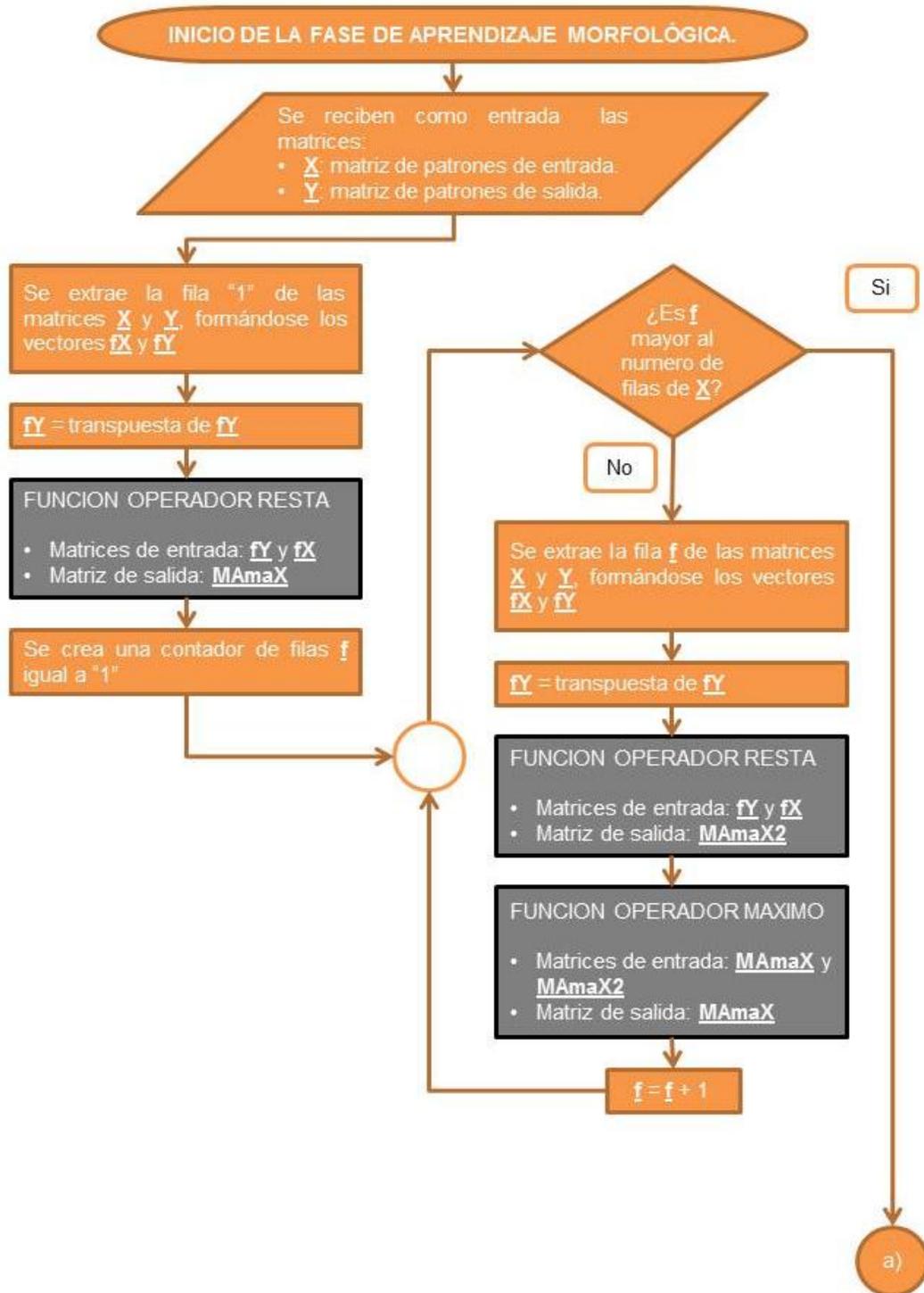
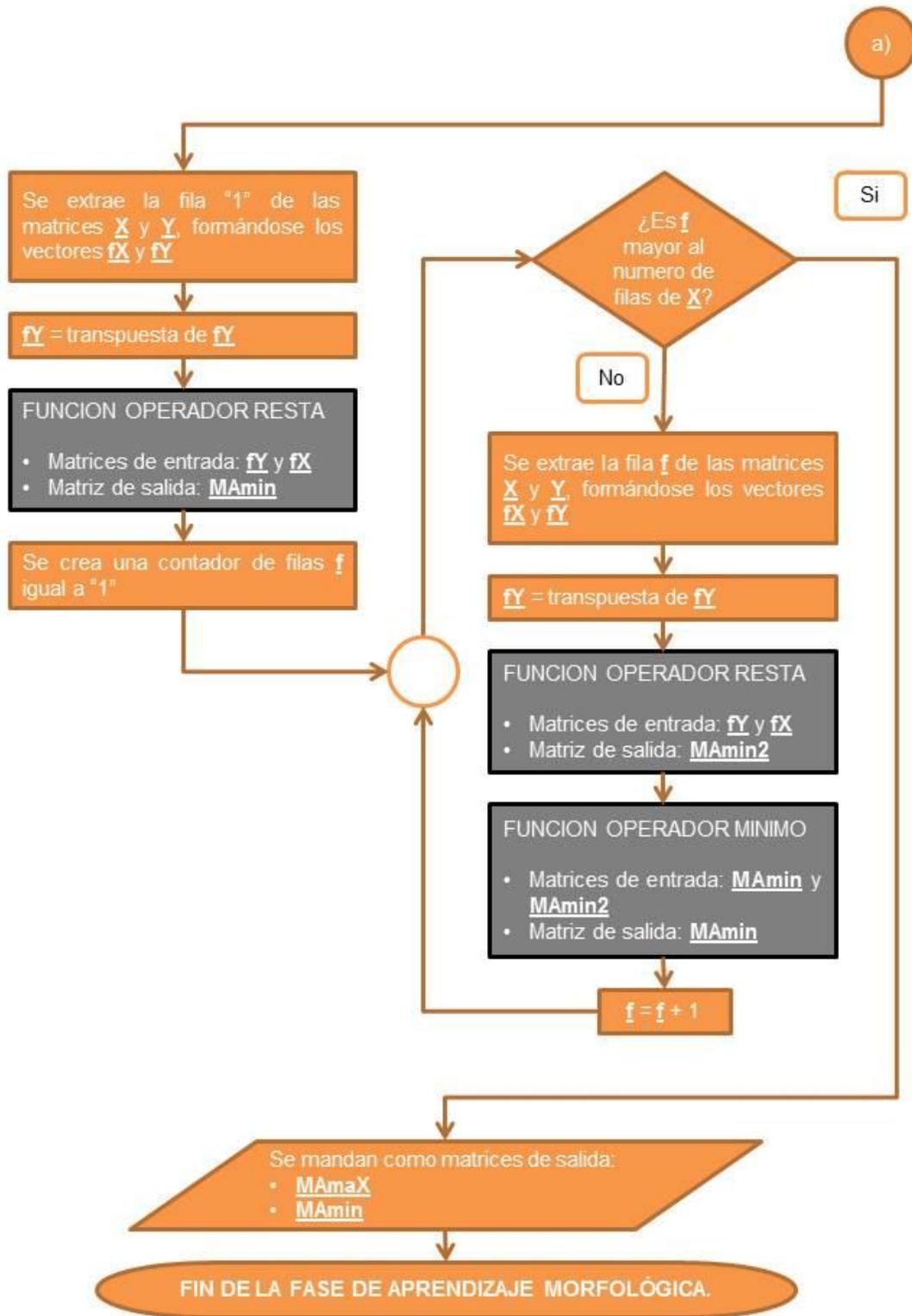


Figura 12: Segunda parte del diagrama de la fase de aprendizaje morfológica.



• OPERADOR RESTA

Usada por la fase de aprendizaje de BAM morfológica realiza una peculiar resta de los elementos de dos vectores (el primero en vertical y el segundo en horizontal), formándose una matriz con los resultados de las restas.

Figura 13: Primera parte del operador resta.

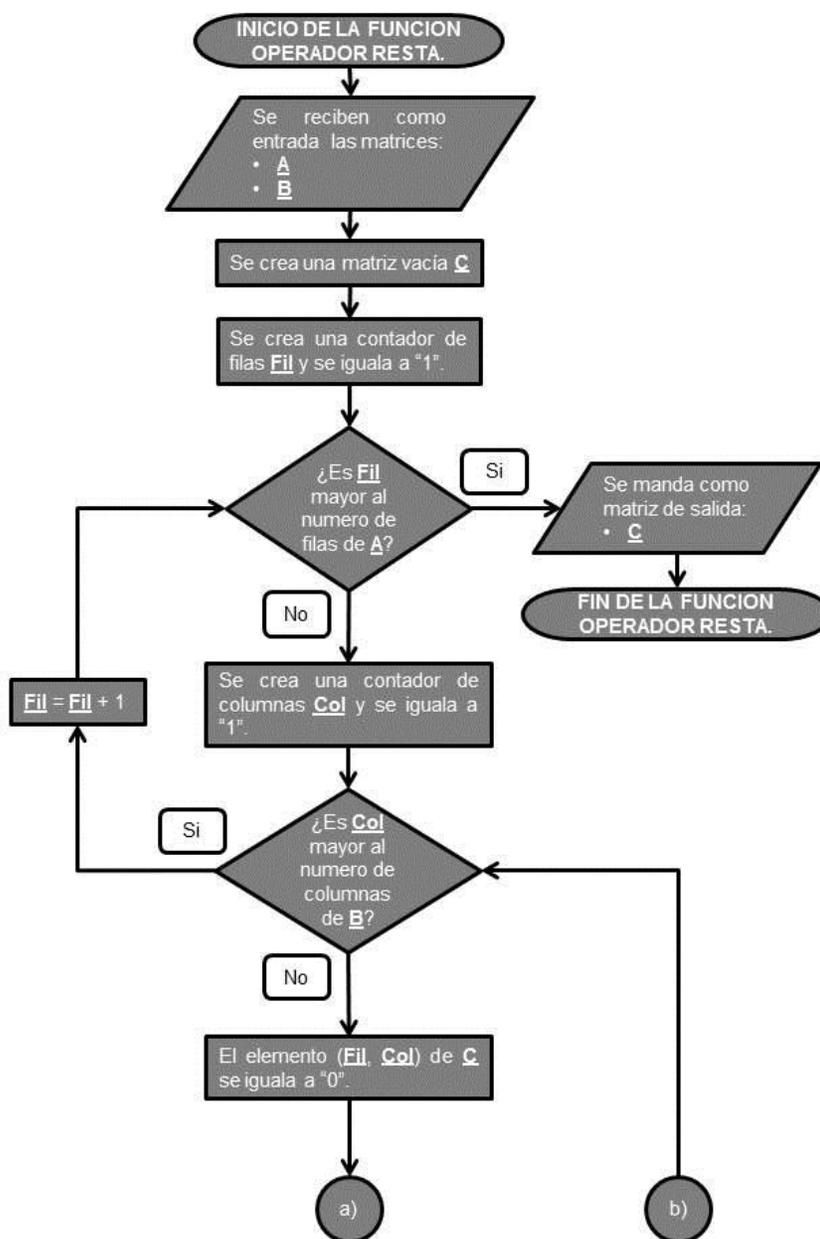
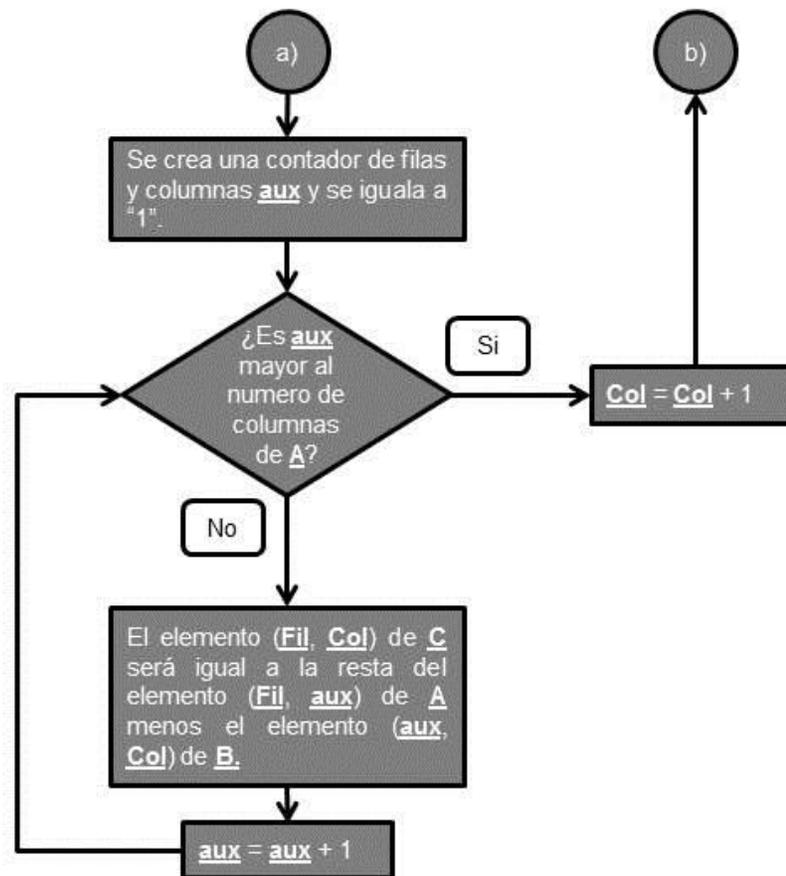


Figura 14: Segunda parte del operador resta.



- **OPERADOR MÁXIMO Y MÍNIMO**

Es un mismo algoritmo que solo cambia en una parte específica. Consiste en comparar los elementos de una matriz con los de otra (solo compara elementos que se encuentren en la misma posición en ambas matrices); formara dos nuevas matrices, la primera conservando el elemento mayor y la segunda el menor.

Figura 15: Primera parte del operador máximo.

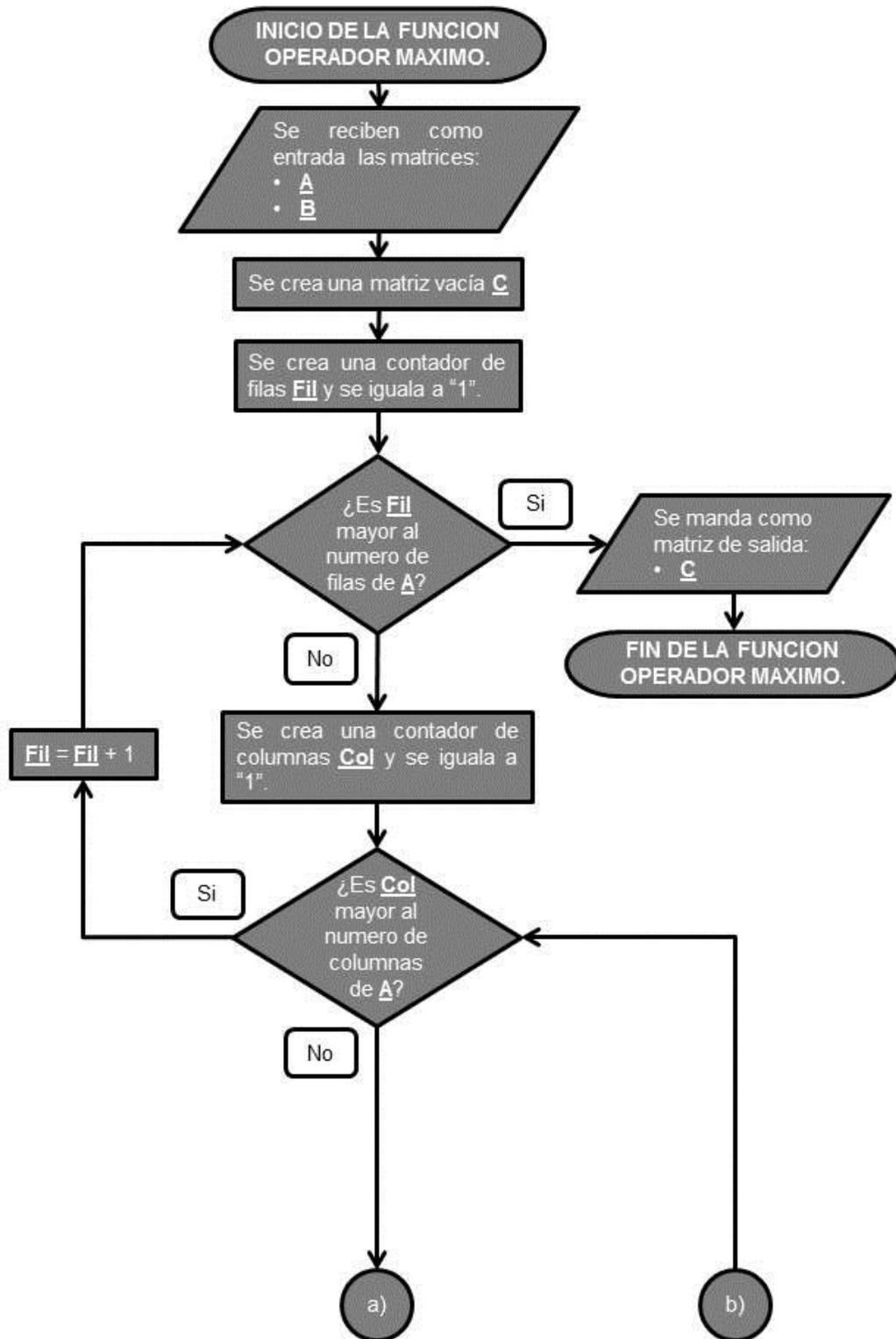


Figura 16: Segunda parte del operador máximo.

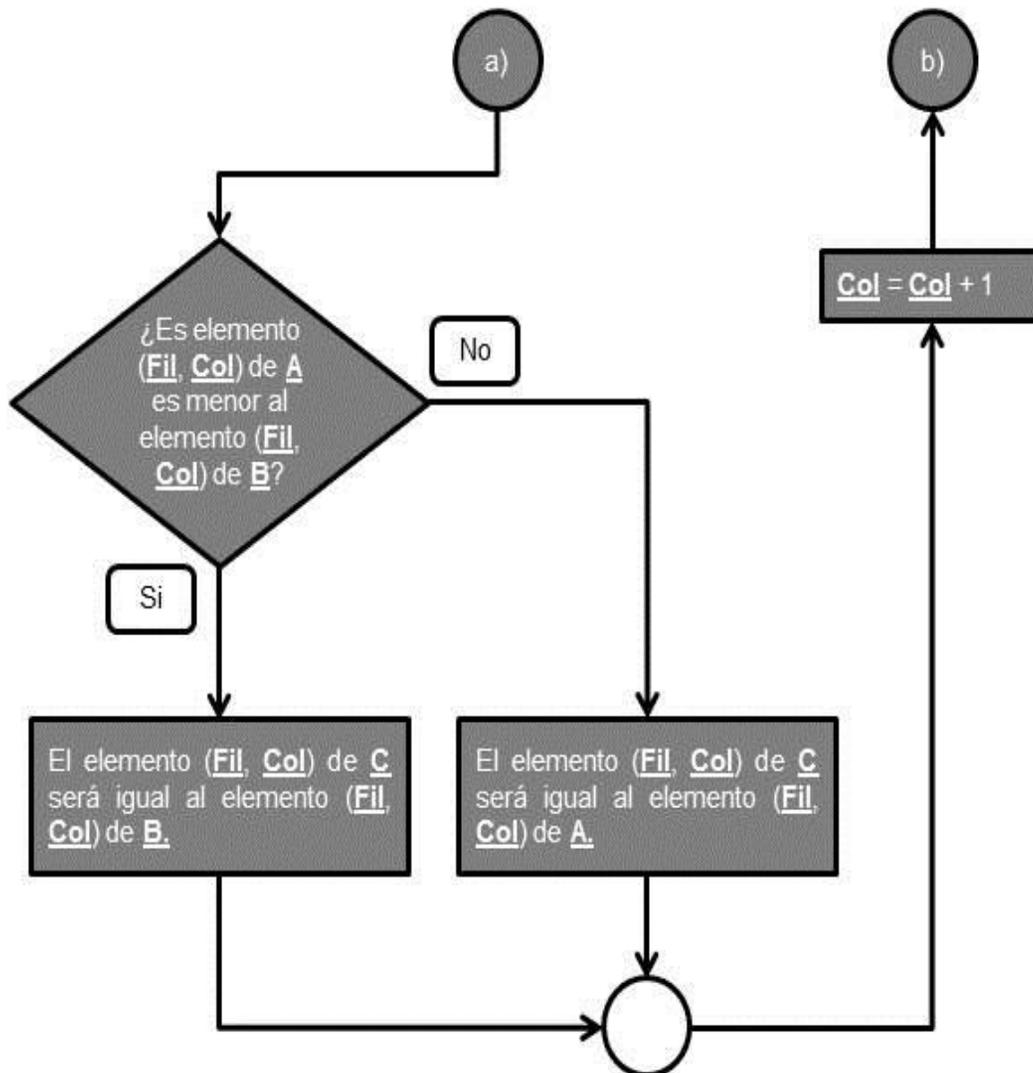


Figura 17: Primera parte del operador mínimo.

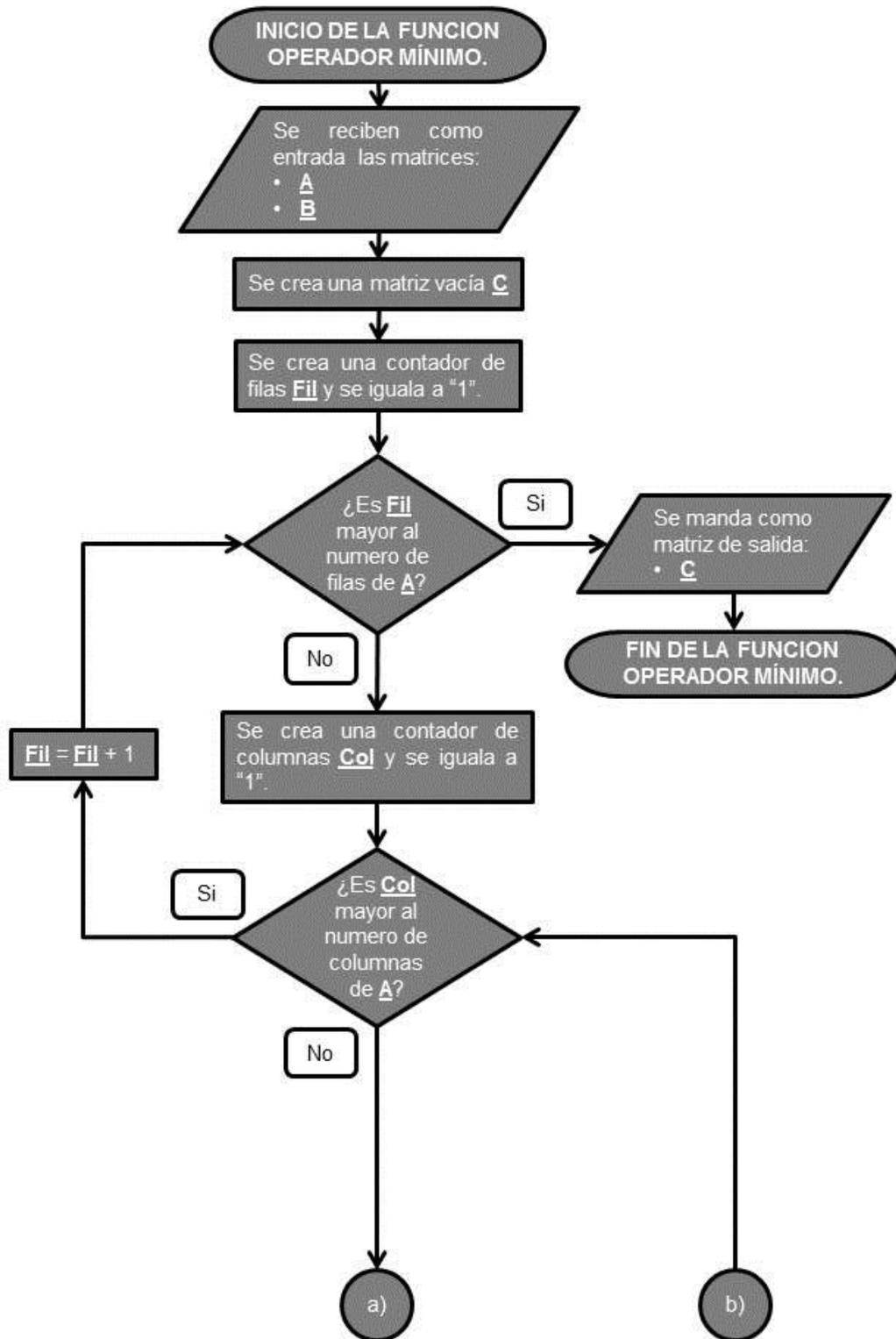
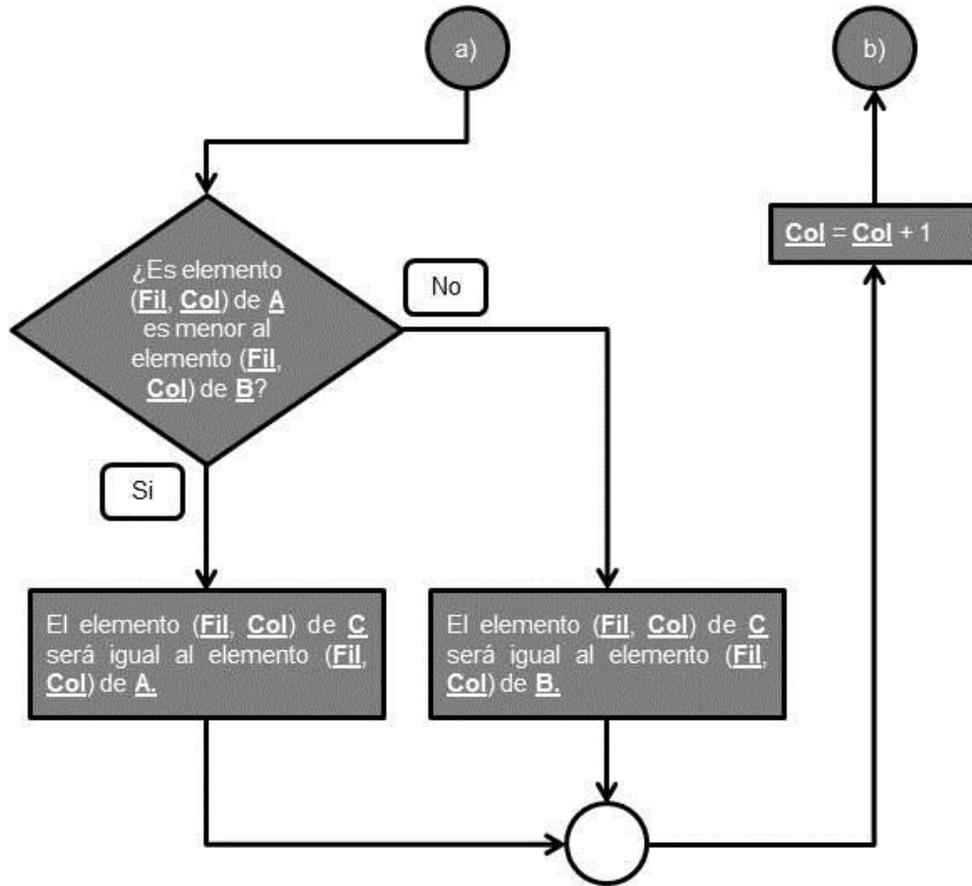


Figura 18: Segunda parte del operador mínimo.



4.2.3 FASES DE RECUPERACIÓN

Después de que el usuario ingresa una palabra o frase a traducir, es necesario procesarla en las diferentes fases de recuperación del sistema. El objetivo es encontrar la traducción correspondiente a la palabra más parecida que se encuentre en la base de datos.

- **FASE DE RECUPERACIÓN DEL DICCIONARIO**

Recibe directamente la palabra o frase que ingreso el usuario desde la interfaz gráfica. Para funcionar debe recibir como parámetros de entrada las memorias asociativas creadas en la fase de aprendizaje.

También tiene la característica de que si en lugar de recibir una palabra recibe el nombre y la extensión de un archivo "txt", abrirá el mismo para traducir las palabras o frases que se encuentren encerradas entre comillas. Las palabras traducidas resultantes de esta operación se mostraran en forma de lista en la pantalla de comandos de Matlab.

Figura 19: Primera parte del diagrama de recuperación del diccionario.

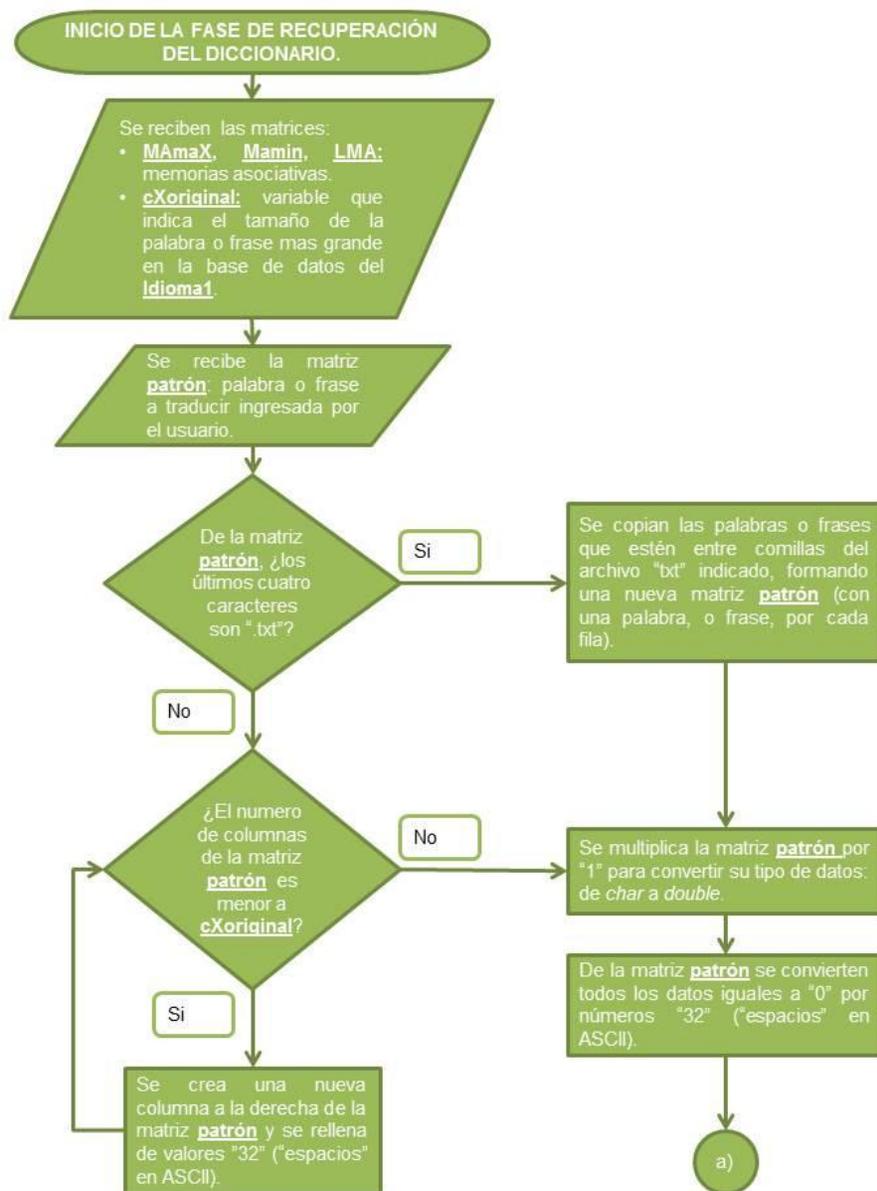
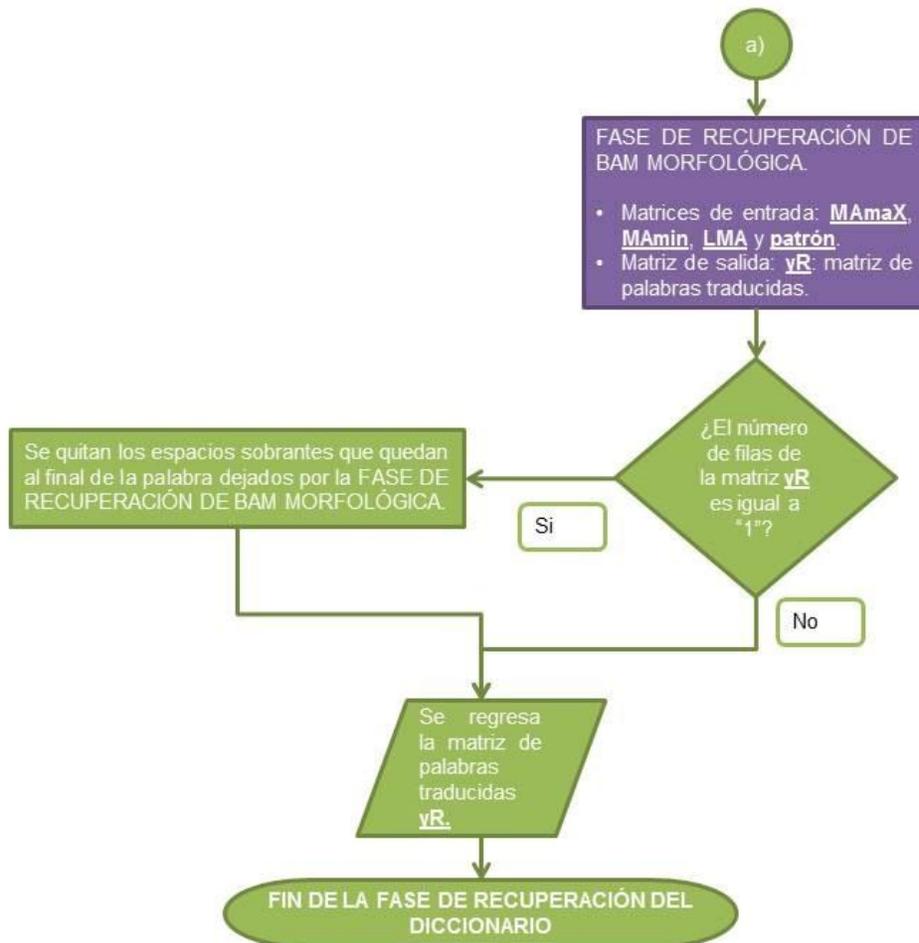


Figura 20: Segunda parte del diagrama de recuperación del diccionario.



- **FASE DE RECUPERACIÓN DE BAM MORFOLÓGICA**

Análoga a la fase de recuperación BAM alfa beta, pero con cambios en el algoritmo necesarios para funcionar con patrones de números enteros. Para empezar, las colas que se concatenan a los patrones son de valores "300" y "-300".

Mientras que en BAM alfa beta se comprobaba si el vector “c” era one hot, aquí cumple esa condición si en el vector “c” hay un valor de “300” y sin importar los demás valores.

De forma similar, el vector “d” cumple como zero hot si uno de sus valores vale “-300” y sin importar los demás valores.

Ya que en BAM alfa beta se manejan valores binarios y operaciones lógicas binarias, en BAM morfológica se hacen operaciones parecidas pero adaptadas a valores enteros.

La negación de un vector se realiza multiplicando este por “-1”.

La operación *AND* se realiza definiendo una tabla de verdad, en la que “X” puede ser cualquier valor entero:

a	B	<i>AND</i>(a, b)
0	0	0
0	X	0
X	0	0
X	X	X

Figura 21: Primera parte del diagrama de la fase de recuperación de BAM morfológica.

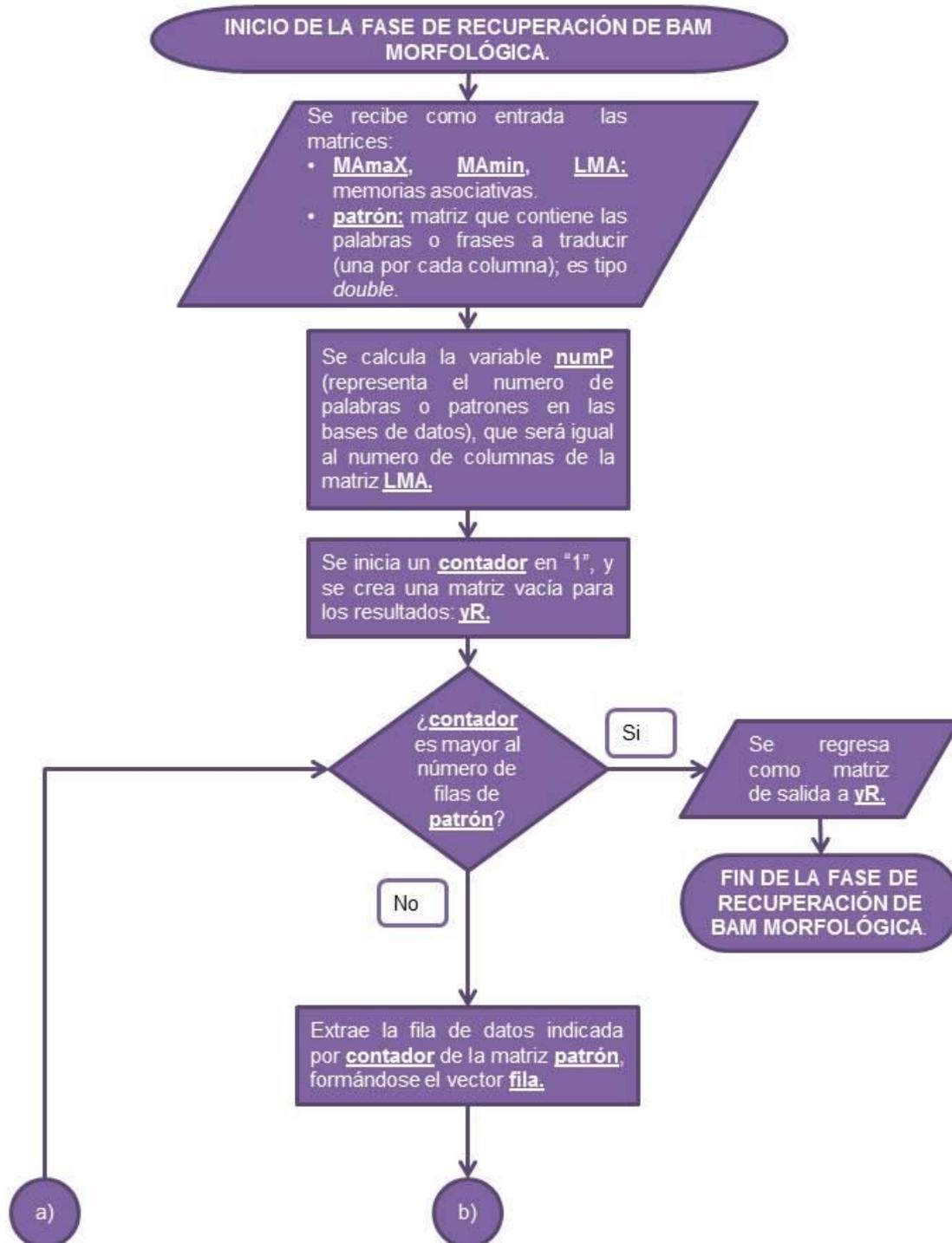


Figura 22: Segunda parte del diagrama de la fase de recuperación de BAM morfológica.

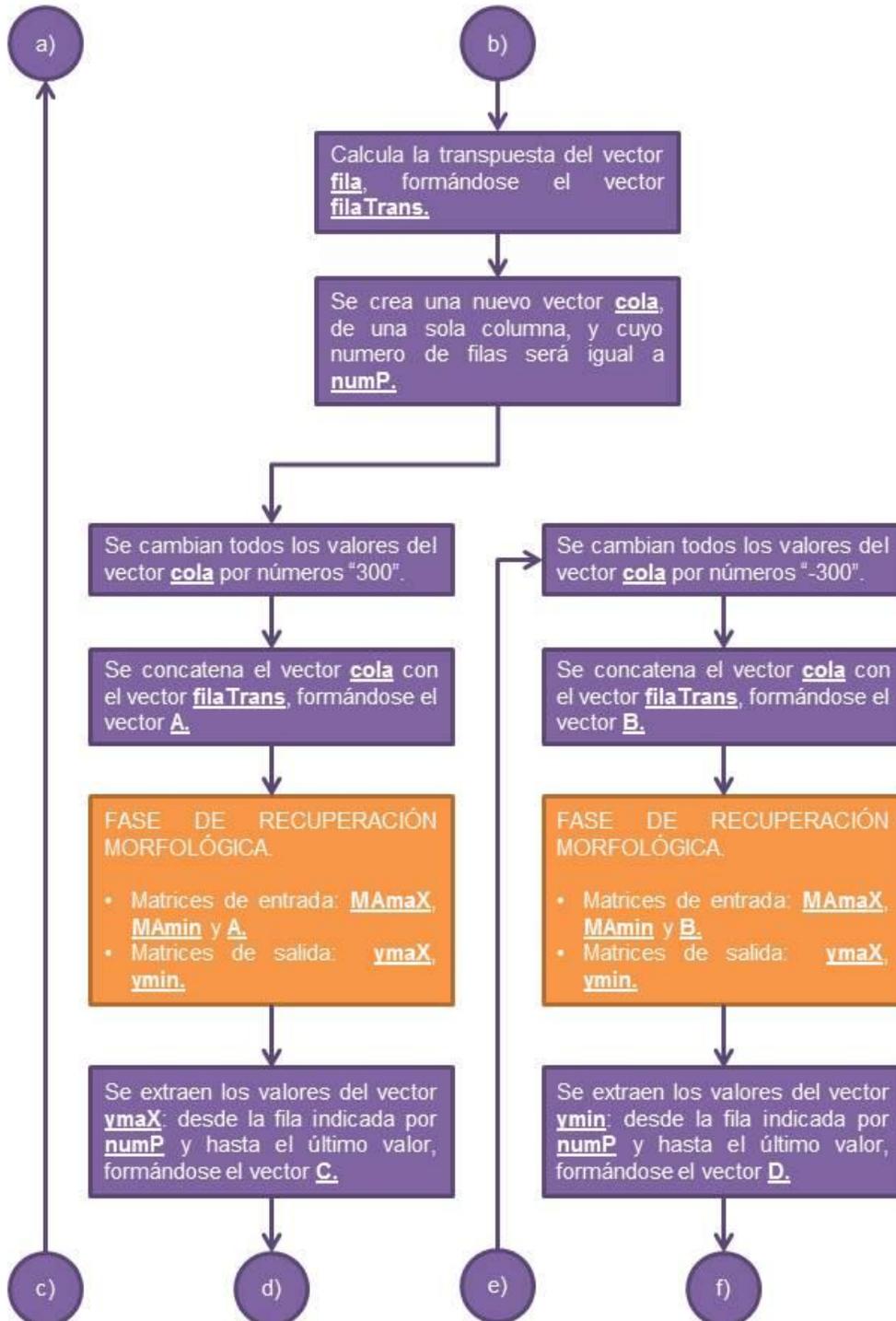
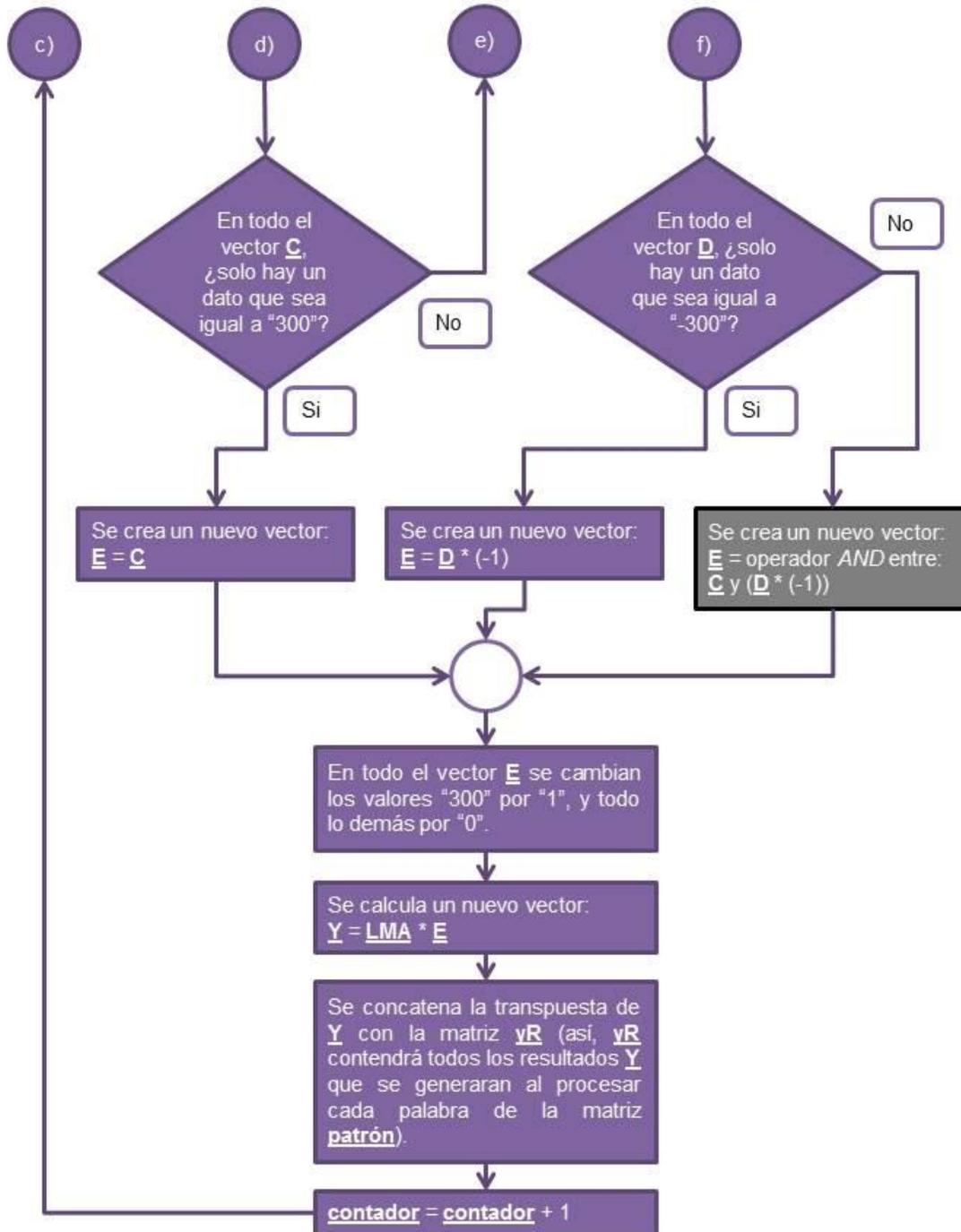


Figura 23: Tercera parte del diagrama de la fase de recuperación de BAM morfológica.



- **FASE DE RECUPERACIÓN MORFOLÓGICA**

Su principal función es la de calcular las matrices **ymáx** y **ymín** para la fase de recuperación BAM morfológica, la cual discrimina alguna de estas dependiendo de si trabaja con el vector “c” o “d”.

Esta función a su vez hace uso de funciones de dilatación, explicadas más adelante.

Figura 24: Diagrama de la fase de recuperación morfológica.



- OPERACIÓN MORFOLÓGICA DE DILATACIÓN

Necesaria para la fase de recuperación morfológica se encarga de calcular las matrices con nombres locales: valoresMáx y valoresMín.

Figura 25: Primera parte del diagrama de la función de operación morfológica de dilatación.

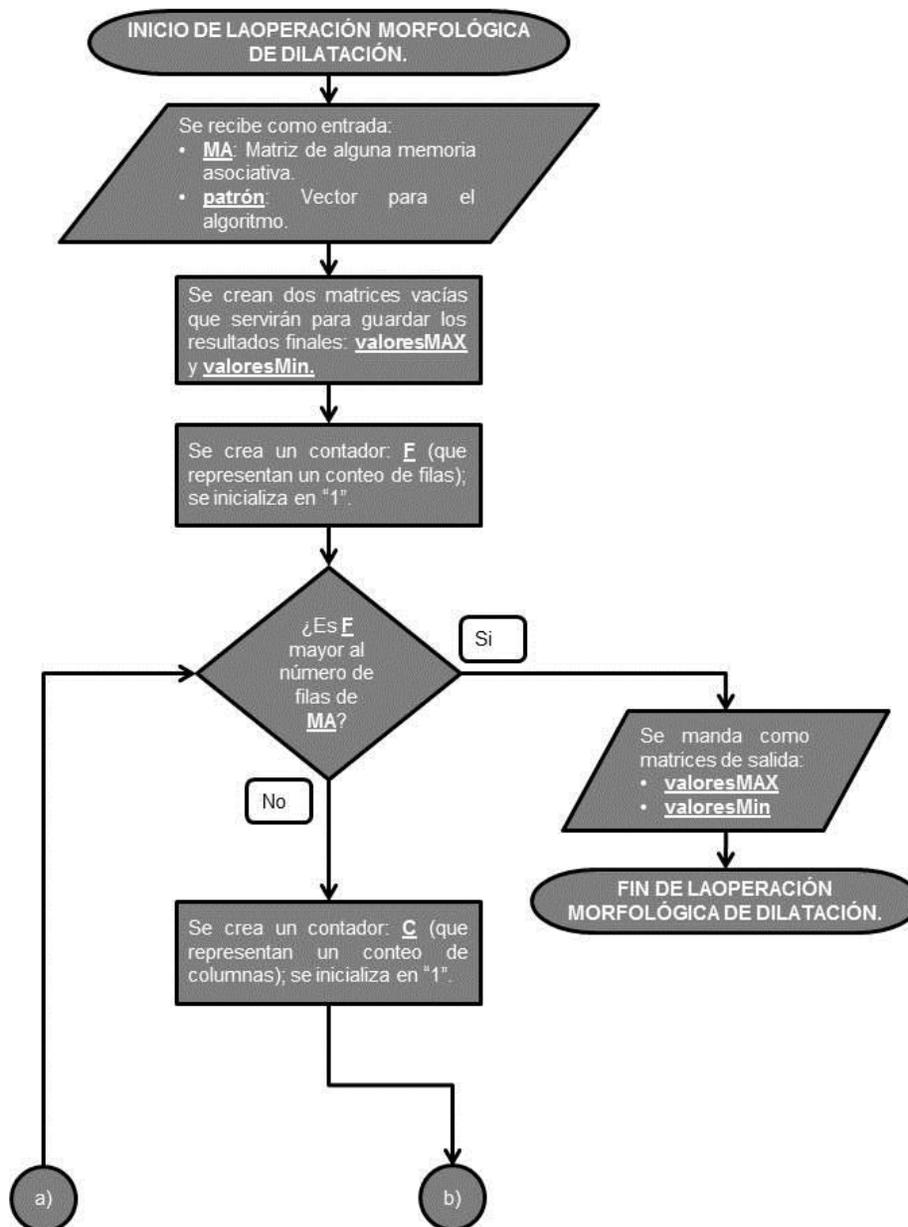


Figura 26: Segunda parte del diagrama de la función de operación morfológica de dilatación.

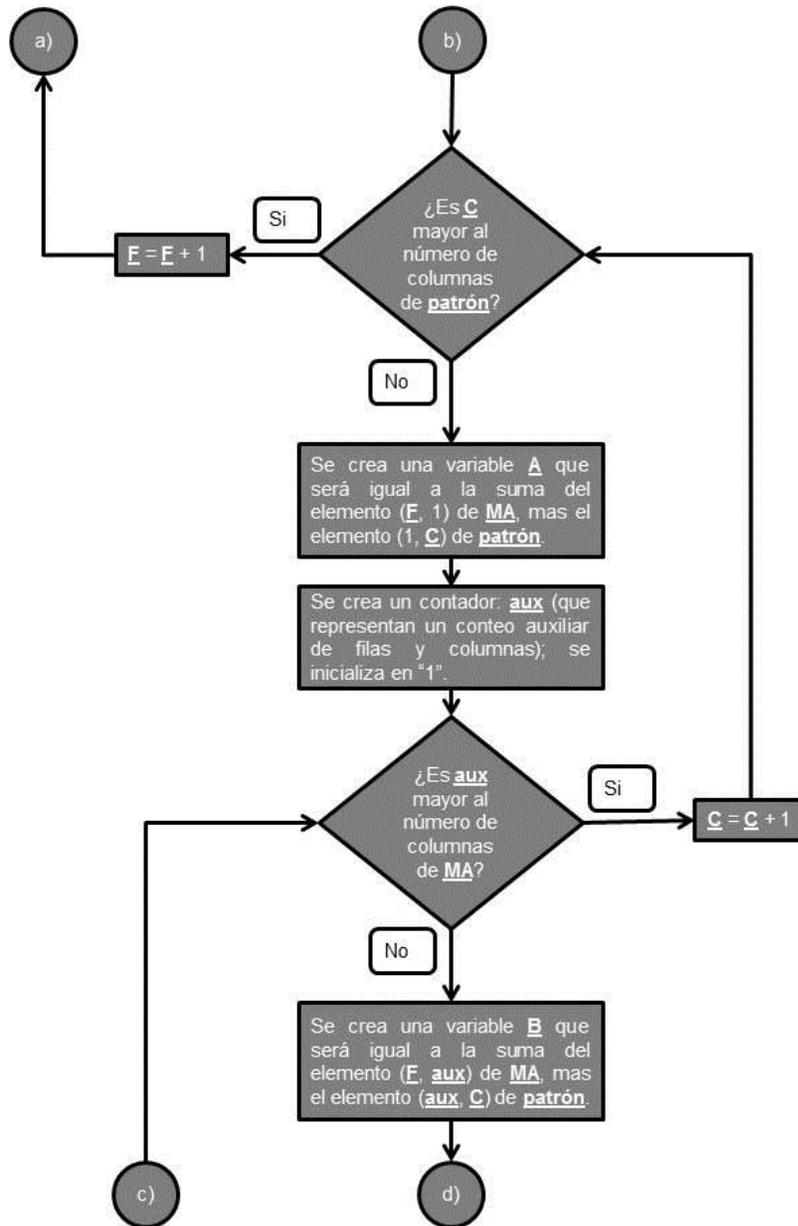
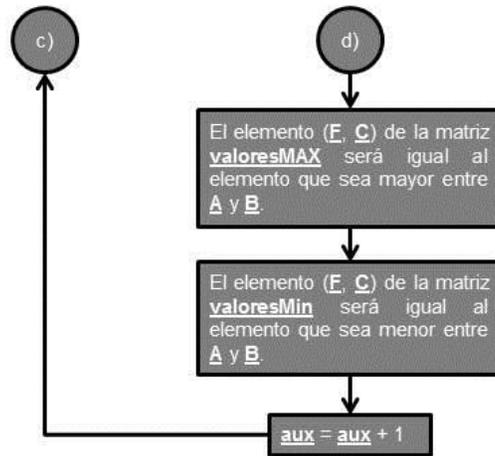


Figura 27: Tercera parte del diagrama de la función de operación morfológica de dilatación.



4.3. EJEMPLO BAM MORFOLÓGICA

4.3.1. FASE DE APRENDIZAJE

Tenemos los siguientes vectores.

$$x^1 = \begin{pmatrix} 58 \\ 70 \\ 25 \end{pmatrix}; y^1 = \begin{pmatrix} 20 \\ 60 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 102 \\ 27 \\ 12 \end{pmatrix}; y^2 = \begin{pmatrix} 10 \\ 30 \end{pmatrix}$$

Como Primer paso concatenamos los vectores One Hot y Zero Hot modificados con valor 300 y -300.

$$x^1 = \begin{pmatrix} 58 \\ 70 \\ 25 \\ 300 \\ 0 \end{pmatrix}$$

$$\bar{x}^1 = \begin{pmatrix} 58 \\ 70 \\ 25 \\ -300 \\ 0 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 102 \\ 27 \\ 12 \\ 0 \\ 300 \end{pmatrix}$$

$$\bar{x}^2 = \begin{pmatrix} 102 \\ 27 \\ 12 \\ 0 \\ -300 \end{pmatrix}$$

Siguiendo los pasos de la sección 4.1 del capítulo

$$x^1 \otimes (-x^1)^t = \begin{pmatrix} 58 - 58 & 58 - 70 & 58 - 25 & 58 - 300 & 58 - 0 \\ 70 - 58 & 70 - 70 & 70 - 25 & 70 - 300 & 70 - 0 \\ 25 - 58 & 25 - 70 & 25 - 25 & 25 - 300 & 25 - 0 \\ 300 - 58 & 300 - 70 & 300 - 25 & 300 - 300 & 300 - 0 \\ 0 - 58 & 0 - 70 & 0 - 25 & 0 - 300 & 0 - 0 \end{pmatrix}$$

$$\bar{x}^1 \otimes (-\bar{x}^1)^t = \begin{pmatrix} 58 - 58 & 58 - 70 & 58 - 25 & 58 + 300 & 58 - 0 \\ 70 - 58 & 70 - 70 & 70 - 25 & 70 + 300 & 70 - 0 \\ 25 - 58 & 25 - 70 & 25 - 25 & 25 + 300 & 25 - 0 \\ -300 - 58 & -300 - 70 & -300 - 25 & -300 + 300 & -300 - 0 \\ 0 - 58 & 0 - 70 & 0 - 25 & 0 + 300 & 0 - 0 \end{pmatrix}$$

$$x^2 \otimes (-x^2)^t = \begin{pmatrix} 102 - 102 & 102 - 27 & 102 - 12 & 102 - 0 & 102 - 300 \\ 27 - 102 & 27 - 27 & 27 - 12 & 27 - 0 & 27 - 300 \\ 12 - 102 & 12 - 27 & 12 - 12 & 12 - 0 & 12 - 300 \\ 0 - 102 & 0 - 27 & 0 - 12 & 0 - 0 & 0 - 300 \\ 300 - 102 & 300 - 27 & 300 - 12 & 300 - 0 & 300 - 300 \end{pmatrix}$$

$$\bar{x}^2 \otimes (-\bar{x}^2)^t = \begin{pmatrix} 102 - 102 & 102 - 27 & 102 - 12 & 102 - 0 & 102 + 300 \\ 27 - 102 & 27 - 27 & 27 - 12 & 27 - 0 & 27 + 300 \\ 12 - 102 & 12 - 27 & 12 - 12 & 12 - 0 & 12 + 300 \\ 0 - 102 & 0 - 27 & 0 - 12 & 0 - 0 & 0 + 300 \\ -300 - 102 & -300 - 27 & -300 - 12 & -300 - 0 & -300 + 300 \end{pmatrix}$$

Obteniendo todos los patrones de entrada modificados pasaremos a formar nuestras Memorias asociativas MÁX y MÍN aplicando la operación morfológica de erosión.

$$V = M = \begin{pmatrix} 0 & 75 & 90 & 102 & 58 \\ 12 & 0 & 45 & 27 & 70 \\ -33 & -15 & 0 & 12 & 25 \\ 242 & 230 & 275 & 0 & 300 \\ 198 & 273 & 288 & 300 & 0 \end{pmatrix}$$

$$\Lambda = W = \begin{pmatrix} 0 & -12 & 33 & 102 & 58 \\ -75 & 0 & 15 & 27 & 70 \\ -90 & -45 & 0 & 12 & 25 \\ -358 & -370 & -325 & 0 & -300 \\ -402 & -327 & -312 & -300 & 0 \end{pmatrix}$$

Una vez obtenidas las Memorias Máximas y Mínimas con los vectores de entrada entonces finalizara esta etapa con los patrones de salida “Y” modificados con el método Linear Associator

Quedando de manera resumida la matriz de asociación:

$$LMAy = \begin{pmatrix} 20 & 10 \\ 60 & 30 \end{pmatrix}$$

4.3.2. FASE DE RECUPERACIÓN

Para la recuperación de “Y” patrón se espera recibir un patrón de entrada “X” con el cual se crearan dos vectores ruidosos que serán representados con las variables A y B dichos vectores serán creados con el patrón “X” y la concatenación de los vectores U y \bar{U} modificados.

$$x^1 = \begin{pmatrix} 58 \\ 70 \\ 25 \end{pmatrix}$$

$$U = \begin{pmatrix} 300 \\ 300 \end{pmatrix}$$

$$\bar{U} = \begin{pmatrix} -300 \\ -300 \end{pmatrix}$$

Quedando expandidos de la siguiente manera:

$$A = \begin{pmatrix} 58 \\ 70 \\ 25 \\ 300 \\ 300 \end{pmatrix}$$

$$B = \begin{pmatrix} 58 \\ 70 \\ 25 \\ -300 \\ -300 \end{pmatrix}$$

Ahora presentamos A a V y B a Λ para obtener dos nuevos vectores, nos enfocaremos en analizar las contracciones de dichos vectores a los que llamaremos vectores C y D que se podría decir que son los vectores U y \bar{U} alterados por la operación de dilatación.

$$M\Delta x = \begin{bmatrix} \bigwedge_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigwedge_{i=1}^n (y_m - x_i + x_i) \end{bmatrix} = y$$

$$W\nabla x = \begin{bmatrix} \bigvee_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigvee_{i=1}^n (y_m - x_i + x_i) \end{bmatrix} = y$$

$$V\Delta A = M\Delta A = \begin{pmatrix} 0 + 58 & 75 + 70 & 90 + 25 & 102 + 300 & 58 + 300 \\ 12 + 58 & 0 + 70 & 45 + 25 & 27 + 300 & 70 + 300 \\ -33 + 58 & -15 + 70 & 0 + 25 & 12 + 300 & 25 + 300 \\ 242 + 58 & 230 + 70 & 275 + 25 & 0 + 300 & 300 + 300 \\ 198 + 58 & 273 + 70 & 288 + 25 & 300 + 300 & 0 + 300 \end{pmatrix}$$

$$V\Delta A = M\Delta A = \begin{pmatrix} 58 & 145 & 115 & 402 & 358 \\ 70 & 70 & 70 & 327 & 370 \\ 25 & 55 & 25 & 312 & 325 \\ 300 & 300 & 300 & 300 & 600 \\ 256 & 343 & 313 & 600 & 300 \end{pmatrix}$$

$$V\Delta A = \begin{pmatrix} 58 \\ 70 \\ 25 \\ 300 \\ 256 \end{pmatrix}$$

Por lo tanto C quedaría

$$C = \begin{pmatrix} 300 \\ 256 \end{pmatrix}$$

Basándonos en el umbral del algoritmo los valores iguales a 300 se cambian a 1 y los demás a 0 por lo tanto.

$$C = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Ya en este último paso aplicamos el algoritmo Linear Associator

$$y = LMAy \cdot C^t = \begin{pmatrix} 20 & 10 \\ 60 & 30 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 20 \\ 60 \end{pmatrix}$$

CAPÍTULO 5. PRUEBAS Y RESULTADOS

En este capítulo se realizarán diferentes pruebas con el prototipo del software. En la sección 5.1 se muestran capturas del prototipo en funcionamiento. Los resultados de las pruebas sobre el prototipo se muestran en la sección 5.2 Finalmente en la sección 5.3 se realizan las conclusiones sobre los objetivos establecidos en base a las pruebas hechas.

5.1. PRESENTACIÓN DEL PROTOTIPO

En esta sección se mostrarán capturas de pantalla del diseño final del prototipo y sus funciones principales.

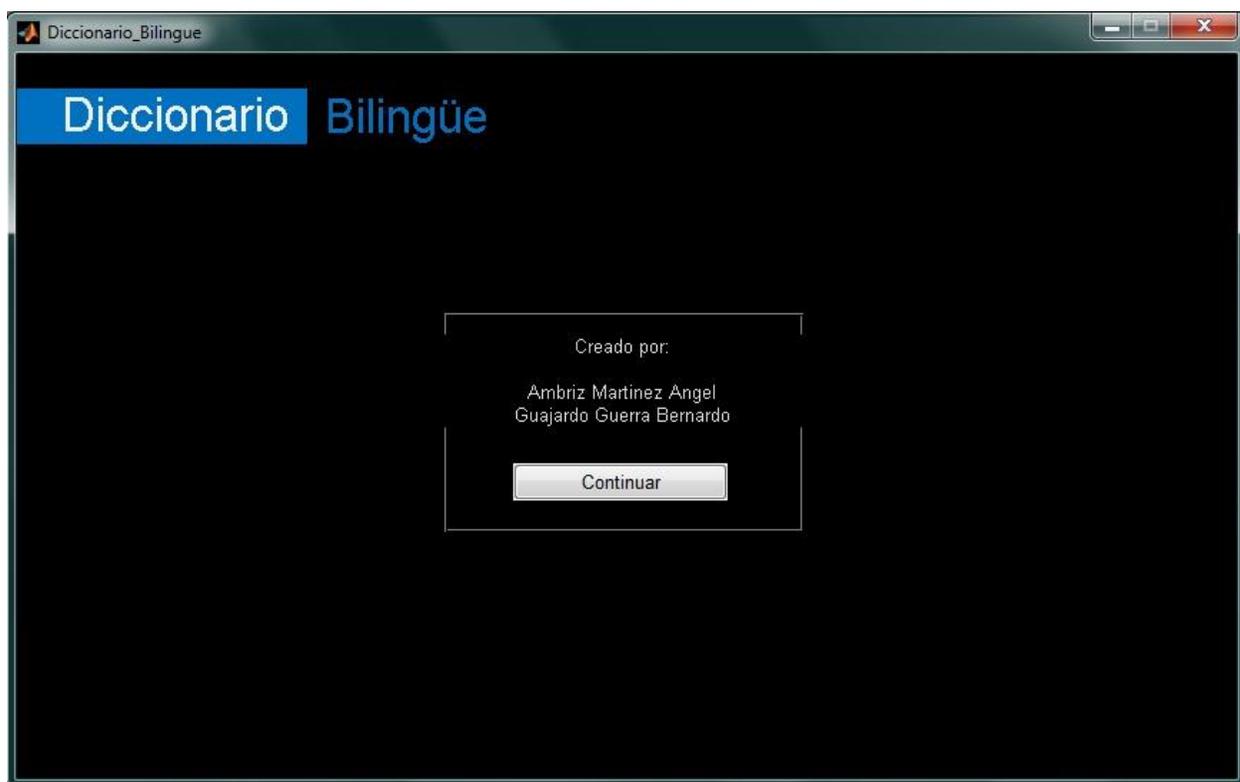


Figura 28: Pantalla inicial. Se muestran los nombres de los autores.

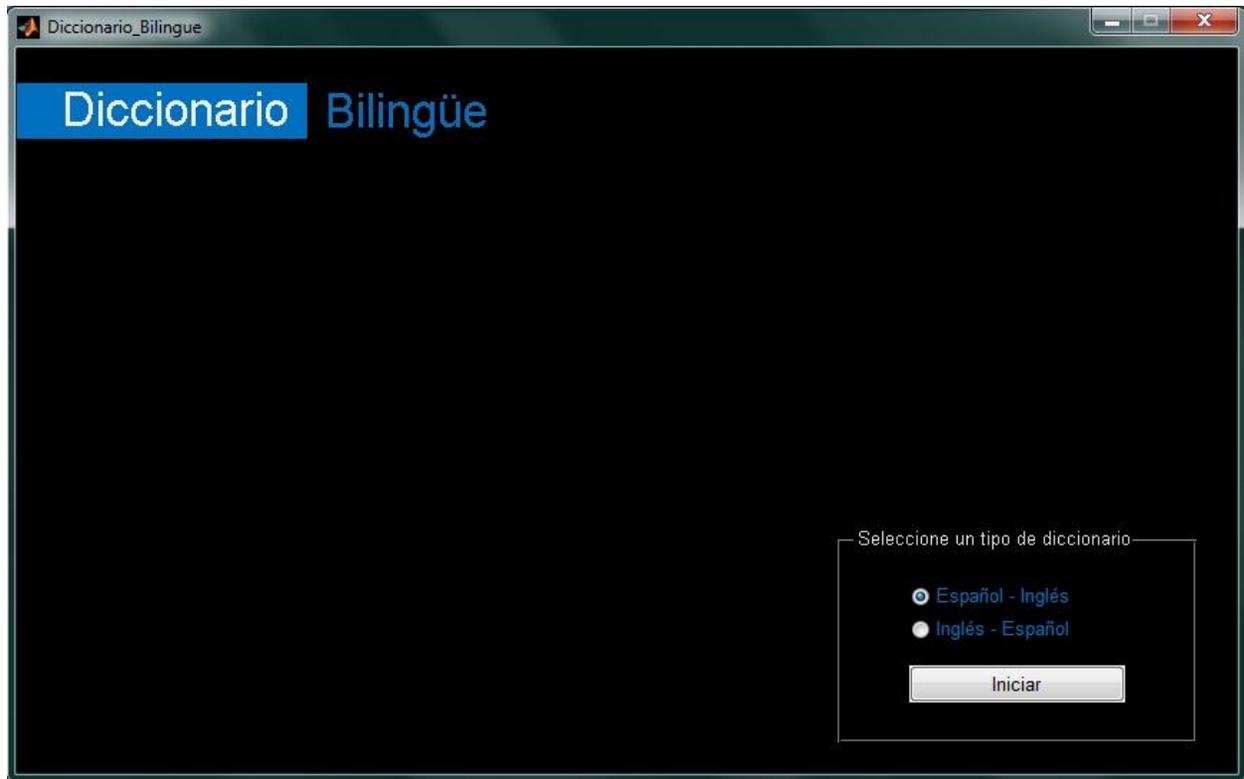


Figura 29: Pantalla para la selección de diccionario inicial.

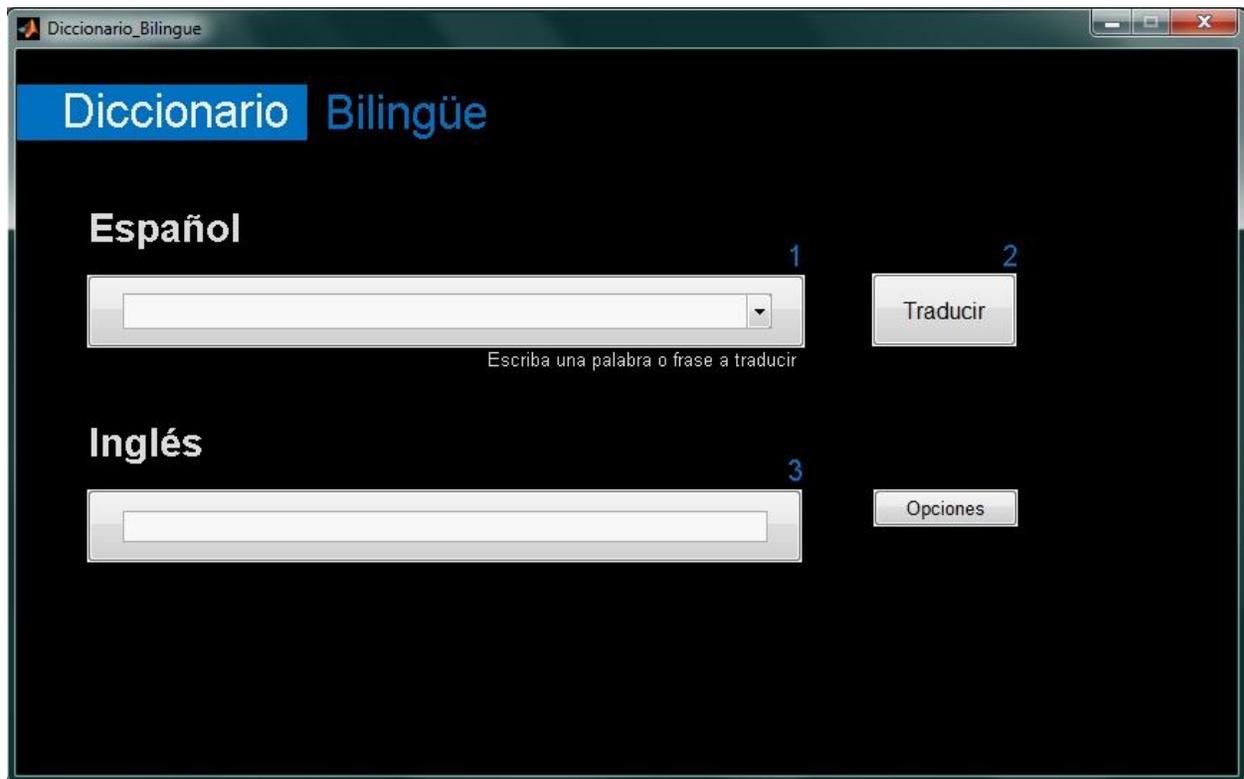


Figura 30: Menú principal para traducir de español a inglés.

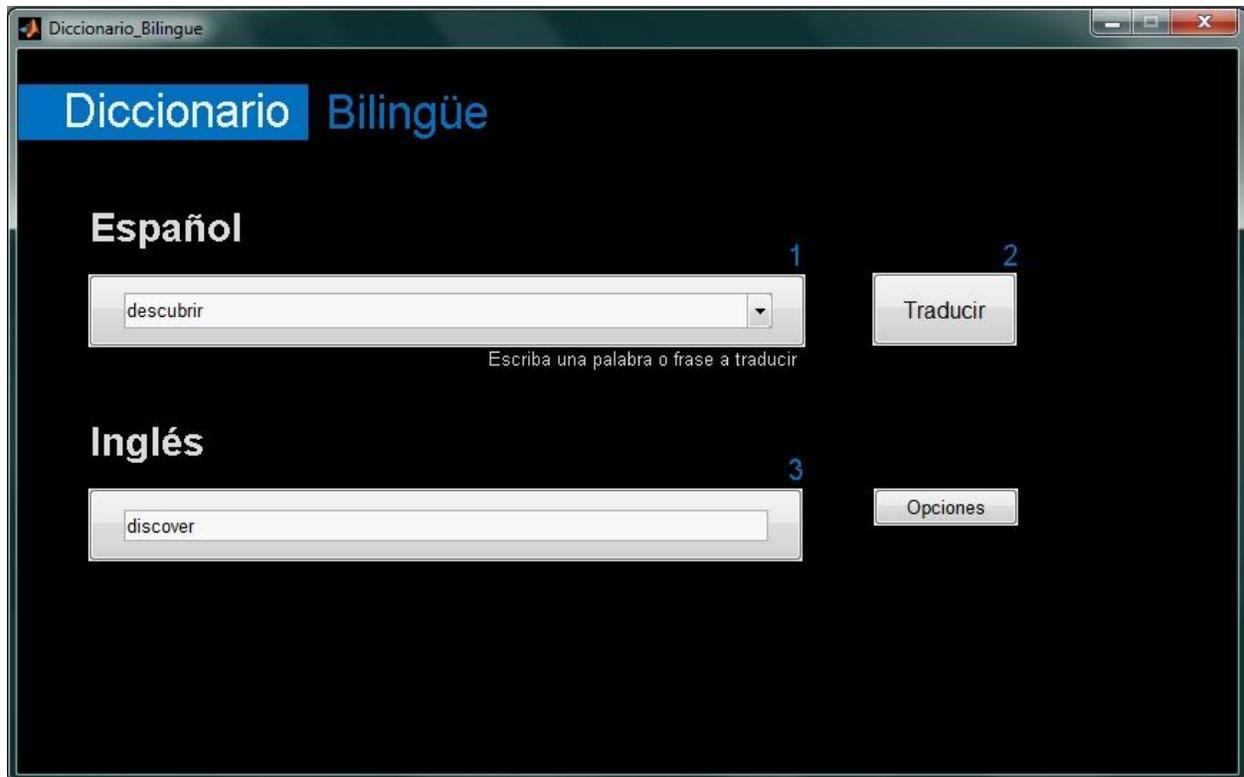
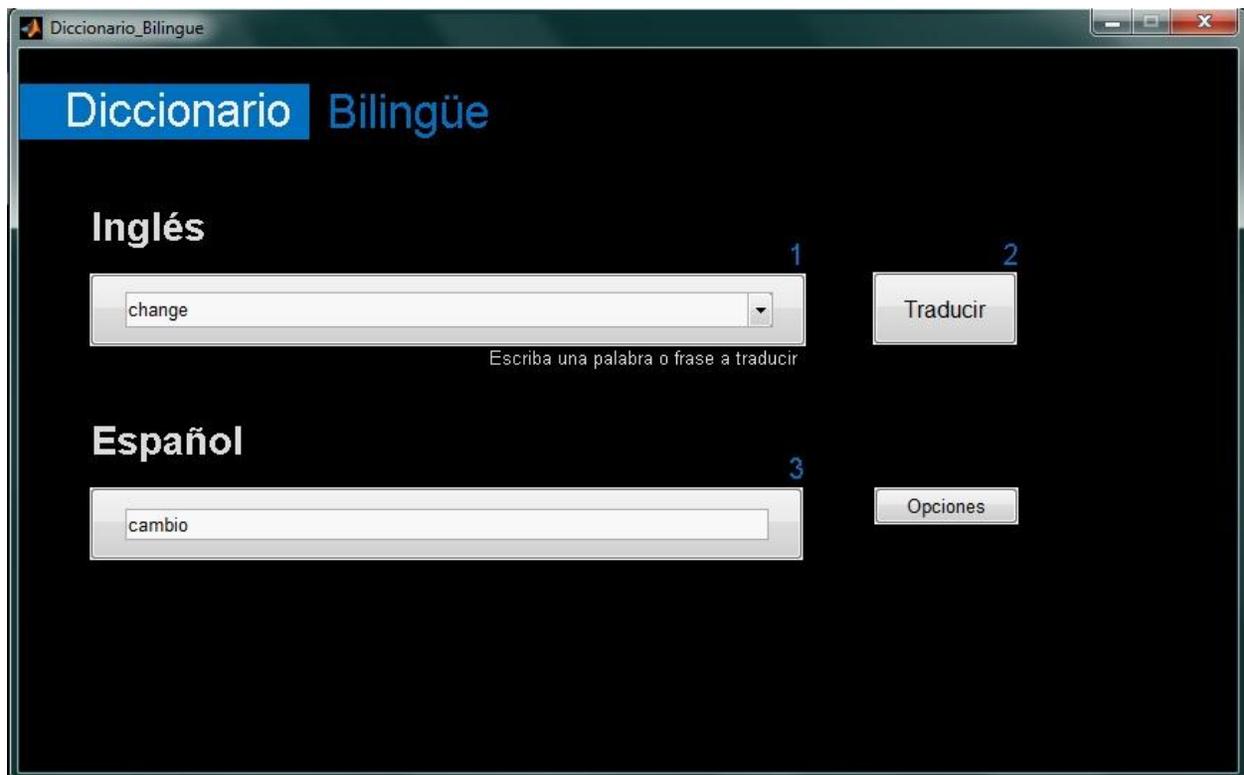


Figura 31: Ejemplo de una traducción de español-inglés.



Ejemplo de una traducción inglés-español.

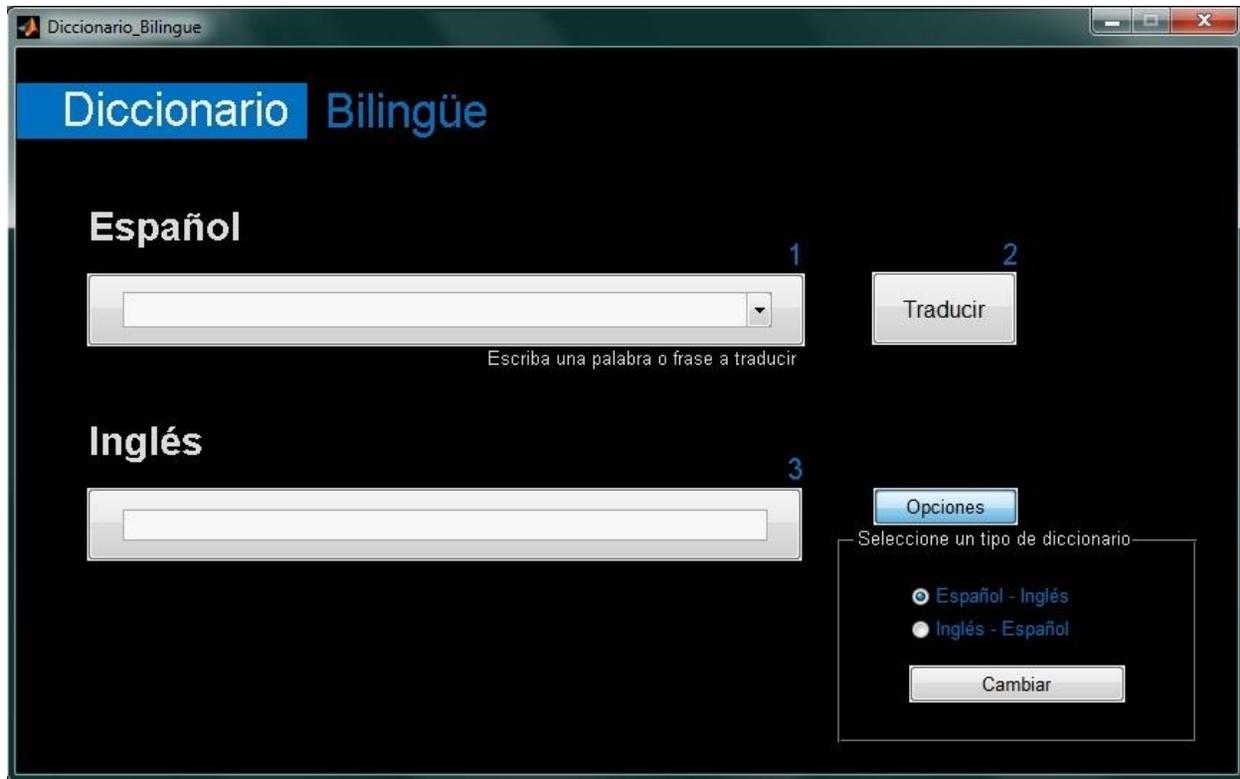


Figura 32: Menú de opciones para cambiar de tipo de diccionario.

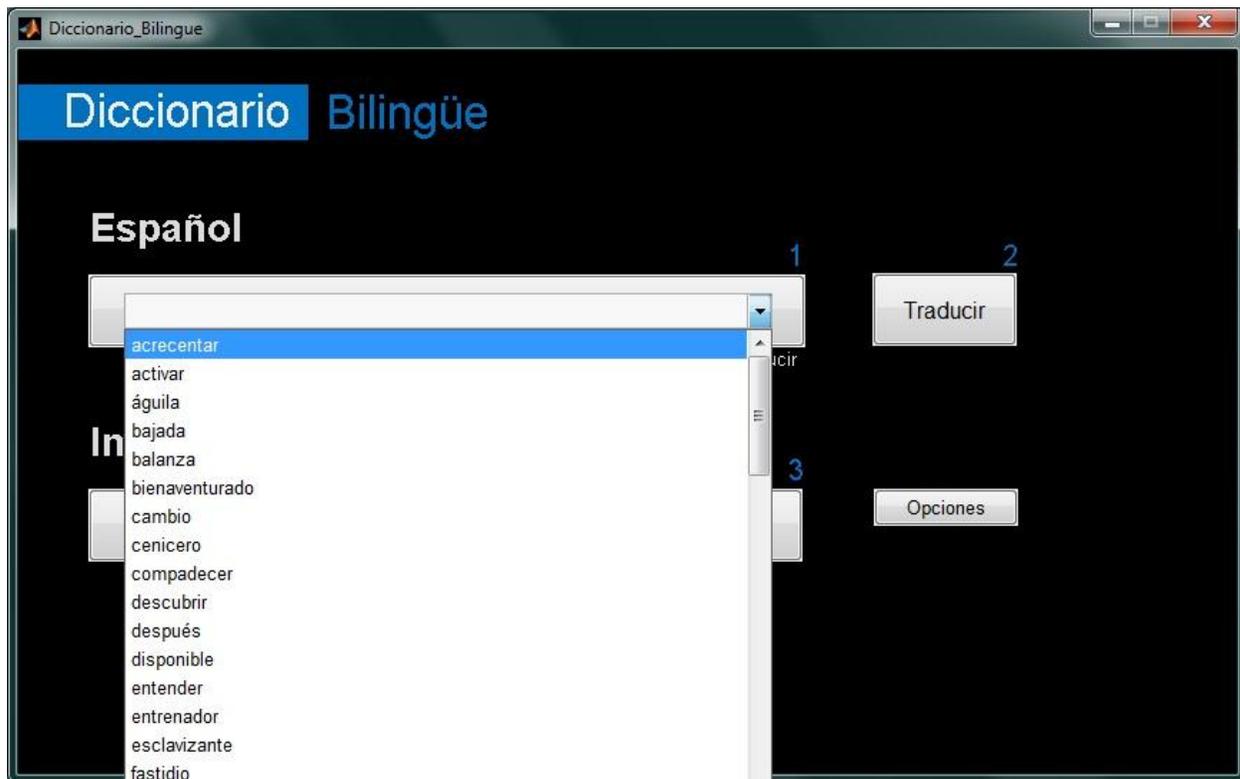


Figura 33: Es posible desplegar la base de datos completa.

5.2. PRUEBAS SOBRE EL PROTOTIPO

Como se ha mencionado anteriormente el software de “Diccionario Bilingüe” desarrollado en este trabajo funciona con el algoritmo BAM morfológica, el cual fue desarrollado para esta aplicación imitando al algoritmo de BAM alfa beta, pero adaptándolo para funcionar con patrones de valores enteros.

A continuación se mostrarán los resultados de pruebas realizadas con el prototipo, con el fin de demostrar su capacidad de recuperación, tiempos de carga y respuesta, y tolerancia al ruido mezclado en diferentes situaciones.

5.2.1 CONSIDERACIONES PREVIAS

El prototipo funciona y será probado bajo las siguientes condiciones:

- Computadora:
 - Sistema operativo Windows 7 Ultimate
 - Microprocesador AMD Dual-Core C-50 a 1 GHz
 - Tarjeta gráfica AMD Raedon HD 6250
 - Memoria RAM de 1 GB DDR3
 - Disco duro de 250 GB
 - Tema AERO activado sin transparencias
 - Modo de alto rendimiento con cargador conectado

- Entorno de desarrollo:
 - Matlab R2010a

- Bases de datos (en español e inglés):
 - 30 palabras
 - 75 palabras y 10 frases
 - 152 palabras y 10 frases

5.2.2 CAPACIDAD DE RECUPERACIÓN

La primera prueba consiste en cargar bases de datos con diferentes cantidades de palabras y contar el número de palabras que son recuperadas (sin ruido).

Tamaño de las bases de datos	Cantidad de palabras recuperadas
30 palabras	30 = 100%
75 palabras y 10 frases	85 = 100%
152 palabras y 10 frases	162 = 100%

5.2.3 TIEMPOS DE CARGA Y RESPUESTA

En esta sección se mide el tiempo que le toma crear todas las memorias asociativas; también se toma el tiempo que tarda en traducir una sola palabra y posteriormente toda la base de datos.

Tamaño de las bases de datos.	Tiempo de carga de las MA (en segundos).	Tiempo de recuperación de una palabra (en segundos).	Tiempo de recuperación de toda la base de datos (en segundos).
30 palabras	1.5 seg. aprox.	Inmediato	1.5 seg. aprox.
75 palabras y 10 frases	9 seg. aprox.	Inmediato	8 seg. aprox.
152 palabras y 10 frases	55 seg. aprox.	Inmediato	39 seg. aprox.

5.2.4 TOLERANCIA AL RUIDO

Se tomaron 5 palabras al azar y se ingresaron con los tipos de ruido descritos a continuación; los resultados fueron:

Palabras	Una letra mas	¿Se recuperó ?	Una letra menos	¿Se recuperó ?	Una letra diferente	¿Se recuperó ?
Descubrir	descubrirr	<No>	descubri	"descent"	descubris	<No>
Luminiscencia	Luminiscencia	<No>	Luminiscencia	<Si>	Luminiscencia	<No>
Hierro	hierroo	<Si>	Hierr	<No>	hierrop	<Si>
Musgo	Musgoo	<No>	Musg	<Si>	musgp	<No>
Siempre	siempree	<No>	siempr	"milk"	siemprf	<No>

5.3. CASOS DE USO

Posibles casos de uso para un algoritmo BAM Morfológica:

Corrección de errores.

Traductor de Idiomas.

5.4. CONCLUSIONES

De forma general se puede decir que el objetivo principal ha sido alcanzado: se logró desarrollar e implementar el algoritmo BAM morfológica sobre la aplicación de un diccionario bilingüe, el cual cumple con los requisitos propuestos, como son el hecho de funcionar con palabras y frases, de soportar errores de escritura por parte del usuario y de leer bases de datos externas al código fuente.

5.4.1 CONCLUSIONES EN BASE A LOS OBJETIVOS ESPECÍFICOS

- Se logró desarrollar e implementar el algoritmo BAM morfológica a partir del BAM alfa beta. También se logró implementar en una aplicación de Matlab.
- De forma satisfactoria, se logró adaptar el código para funcionar con palabras.
- La aplicación de “Diccionario Bilingüe” es capaz de obtener listas de palabras para asociar, desde bases de datos externas, siempre y cuando cumplan con las características necesarias: que dos bases de datos a asociar tengan el mismo número de palabras o frases, además de que deberán estar escritas entre comillas.
- Se determinó que las bases de datos deberían de tener alrededor de 85 palabras para que el tiempo de carga del “Diccionario Bilingüe” sea relativamente corto.
- Si bien no se estableció una medida exacta en cuanto al nivel de tolerancia de errores por parte del usuario, se concluyó que la cantidad y el tipo de errores que puede superar la aplicación es aceptable pero no ideal: no todas las palabras soportan los mismos tipos de errores. Al menos por el nivel de alcance de este proyecto (que no pretende desarrollar una aplicación comercial) se puede decir que el nivel de tolerancia al error es adecuada.

REFERENCIAS

- [1] Joseba Abaitua. 2001. Introducción a la traducción automática (en 10 horas). Grupo DELi, Universidad de Deusto. Victoria López. 2002. Posibilidades y realidades de la Traducción Automática traducción.rediris.es | Rediris.
- [2] Yáñez-Márquez, C. (2002), Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios, Tesis de Doctorado, IPN Centro de Investigación en Computación, México.
- [3] Acevedo-Mosqueda, M.E. (2006), Memorias Asociativas Bidireccionales Alfa-Beta, Tesis de Doctorado, IPN Centro de Investigación en Computación, México.
- [4] Díaz de León, J.L. & Yáñez, C(1999). Memorias asociativas con respuesta perfecta y capacidad infinita, Memoria de TAINA '99, México, D.F. 23-38.
- [5] Yáñez-Márquez, C. y J.L, Díaz-de-León Santiago. "Memorias Autoasociativas Morfológicas max: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones". IT-175, serie Azul, 970-36-0035-2, CIC-IPN, México 2003.
- [6] Yáñez-Márquez, C. y J.L, Díaz-de-León Santiago. "Memorias Autoasociativas Morfológicas min: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones". IT-177, serie Azul, 970-36-0036-0, CIC-IPN, México 2003.
- [7] Díaz-de-León Santiago, J.L, y C. Yáñez-Márquez, "Memorias Heteroasociativas Morfológicas max: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones", IT-174, Serie Azul, ISBN 970-36-0033-6, CIC-IPN, México, 2003.
- [8] Díaz-de-León Santiago, J.L. y C. Yáñez-Márquez, "Memorias Heteroasociativas Morfológicas min: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones", IT-176, Serie Azul, ISBN 970-36-0034-4, CIC-IPN, México, 2003.
- [9] http://es.wikipedia.org/wiki/Google_Translate.
- [10] <http://www.um.es/docencia/barzana/TEI/Informatica-Aplicada-a-la-Traduccion-Memorias-de-Traduccion.html>.

ANEXO

EJEMPLO NUMERICO DEL ALGORITMO

Se tienen 3 palabras con su traducción en inglés

- vaca → cow
- caja → box
- niño → boy

Transformamos dichas palabras con el código ASCII

- vaca [118, 97, 99, 97] → cow [99, 111, 119]
- caja [99, 97, 106, 97] → box [98, 111, 120]
- niño [110, 105, 164, 111] → boy [98, 111, 121]

Nota: En caso de ocupar una palabra con una cantidad mayor de letras que las demás nos llevaría a rellenar las otras palabras con el número 32, (equivalente en ASCII a un espacio) tantas letras necesitemos para que cada una de las palabras que vayamos a ocupar sean de la misma dimensión.

FASE DE APRENDIZAJE

A continuación trabajamos con dichos vectores de entrada y salida:

Entradas

- $x^1 = [118, 97, 99, 97]$
- $x^2 = [99, 97, 106, 97]$
- $x^3 = [110, 105, 164, 111]$

Salidas

- $y^1 = [99, 111, 119]$
- $y^2 = [98, 111, 120]$
- $y^3 = [98, 111, 121]$

Se concatenan a los vectores X los One Hot y Zero Hot

Vectores One Hot Modificados

$$\begin{bmatrix} 300 & 0 & 0 \\ 0 & 300 & 0 \\ 0 & 0 & 300 \end{bmatrix}$$

- $x^1 = [118, 97, 99, 97, 300, 0, 0]$
- $x^2 = [99, 97, 106, 97, 0, 300, 0]$
- $x^3 = [110, 105, 164, 111, 0, 0, 300]$

Vectores Zero Hot Modificados

$$\begin{bmatrix} -300 & 0 & 0 \\ 0 & -300 & 0 \\ 0 & 0 & -300 \end{bmatrix}$$

- $\bar{x}^1 = [118, 97, 99, 97 - 300, 0, 0]$
- $\bar{x}^2 = [99, 97, 106, 97, 0, -300, 0]$
- $\bar{x}^3 = [110, 105, 164, 111, 0, 0, -300]$

El siguiente paso es realizar la autoasociación de los vectores X

$$x^1 \otimes (-x^1)^t = \begin{bmatrix} 118 \\ 97 \\ 99 \\ 97 \\ 300 \\ 0 \\ 0 \end{bmatrix} \otimes -[118, 97, 99, 97, 300, 0, 0]$$

$$= \begin{bmatrix} 118 - 118 & 118 - 97 & 118 - 99 & 118 - 97 & 118 - 300 & 118 - 0 & 118 - 0 \\ 97 - 118 & 97 - 97 & 97 - 99 & 97 - 97 & 97 - 300 & 97 - 0 & 97 - 0 \\ 99 - 118 & 99 - 97 & 99 - 99 & 99 - 97 & 99 - 300 & 99 - 0 & 99 - 0 \\ 97 - 118 & 97 - 97 & 97 - 99 & 97 - 97 & 97 - 300 & 97 - 0 & 97 - 0 \\ 300 - 118 & 300 - 97 & 300 - 99 & 300 - 97 & 300 - 300 & 300 - 0 & 300 - 0 \\ 0 - 118 & 0 - 97 & 0 - 99 & 0 - 97 & 0 - 300 & 0 - 0 & 0 - 0 \\ 0 - 118 & 0 - 97 & 0 - 99 & 0 - 97 & 0 - 300 & 0 - 0 & 0 - 0 \end{bmatrix}$$

La memoria resultante del patrón x^1

$$= \begin{bmatrix} 0 & 21 & 19 & 21 & -182 & 118 & 118 \\ -21 & 0 & -2 & 0 & -203 & 97 & 97 \\ -19 & 2 & 0 & 2 & -201 & 99 & 99 \\ -21 & 0 & -2 & 0 & -203 & 97 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ -118 & -97 & -99 & -97 & -300 & 0 & 0 \\ -118 & -97 & -99 & -97 & -300 & 0 & 0 \end{bmatrix}$$

De la misma manera se obtienen los otros 2 patrones

$$x^2 \otimes (-x^2)^t = \begin{bmatrix} 99 \\ 97 \\ 106 \\ 97 \\ 0 \\ 300 \\ 0 \end{bmatrix} \otimes - [99, 97, 106, 97, 0, 300, 0]$$

$$x^2 = \begin{bmatrix} 0 & 2 & -7 & 2 & 99 & -201 & 99 \\ -2 & 0 & -9 & 0 & 97 & -203 & 97 \\ 7 & 9 & 0 & 9 & 106 & -194 & 106 \\ -2 & 0 & -9 & 0 & 97 & -203 & 97 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \\ 201 & 203 & 194 & 203 & -300 & 0 & 300 \\ -99 & -97 & -106 & -97 & 0 & -300 & 0 \end{bmatrix}$$

$$x^3 \otimes (-x^3)^t = \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ 0 \\ 0 \\ 300 \end{bmatrix} \otimes - [110, 105, 164, 111, 0, 0, 300]$$

$$x^3 = \begin{bmatrix} 0 & 5 & -54 & -1 & 110 & 110 & -190 \\ -5 & 0 & -59 & -6 & 105 & 105 & -195 \\ 54 & 59 & 0 & 53 & 164 & 164 & -136 \\ 1 & 6 & -53 & 0 & 111 & 111 & -189 \\ -110 & -105 & -164 & -111 & 0 & 0 & -300 \\ -110 & -105 & -164 & -111 & 0 & 0 & -300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix}$$

Aplicando la operación Morfológica de Erosión obtenemos la Memoria MÁX

$$V = M = \bigvee_{\mu=1}^p (x^\mu \otimes (-x^\mu)^t)$$

Se comparan cada uno de los parámetros de las memorias para obtener el de mayor valor.

$$M = \begin{bmatrix} 0 & 21 & 19 & 21 & -182 & 118 & 118 \\ -21 & 0 & -2 & 0 & -203 & 97 & 97 \\ -19 & 2 & 0 & 2 & -201 & 99 & 99 \\ -21 & 0 & -2 & 0 & -203 & 97 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ -118 & -97 & -99 & -97 & -300 & 0 & 0 \\ -118 & -97 & -99 & -97 & -300 & 0 & 0 \end{bmatrix} V = \begin{bmatrix} 0 & 2 & -7 & 2 & 99 & -201 & 99 \\ -2 & 0 & -9 & 0 & 97 & -203 & 97 \\ 7 & 9 & 0 & 9 & 106 & -194 & 106 \\ -2 & 0 & -9 & 0 & 97 & -203 & 97 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \\ 201 & 203 & 194 & 203 & -300 & 0 & 300 \\ -99 & -97 & -106 & -97 & 0 & -300 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 5 & -54 & -1 & 110 & 110 & -190 \\ -5 & 0 & -59 & -6 & 105 & 105 & -195 \\ 54 & 59 & 0 & 53 & 164 & 164 & -136 \\ 1 & 6 & -53 & 0 & 111 & 111 & -189 \\ -110 & -105 & -164 & -111 & 0 & 0 & -300 \\ -110 & -105 & -164 & -111 & 0 & 0 & -300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 21 & 19 & 21 & 110 & 118 & 118 \\ -2 & 0 & -2 & 0 & 105 & 105 & 97 \\ 54 & 59 & 0 & 53 & 164 & 164 & 106 \\ 1 & 6 & -2 & 0 & 111 & 111 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ 201 & 203 & 194 & 203 & 0 & 0 & 300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix}$$

Ya teniendo la memoria asociativa MÁX pasamos a realizar los mismos pasos pero con la \bar{X}

$$\bar{x}^1 \otimes (-\bar{x}^1)^t = \begin{bmatrix} 118 \\ 97 \\ 99 \\ 97 \\ -300 \\ 0 \\ 0 \end{bmatrix} \otimes [118, 97, 99, 97, -300, 0, 0]$$

$$\bar{x}^1 = \begin{bmatrix} 0 & 21 & 19 & 21 & 418 & 118 & 118 \\ -21 & 0 & -2 & 0 & 397 & 97 & 97 \\ -19 & 2 & 0 & 2 & 399 & 99 & 99 \\ -21 & 0 & -2 & 0 & 397 & 97 & 97 \\ -418 & -397 & -399 & -397 & 0 & -300 & -300 \\ -118 & -97 & -99 & -97 & 300 & 0 & 0 \\ -118 & -97 & -99 & -97 & 300 & 0 & 0 \end{bmatrix}$$

$$\bar{x}^2 \otimes (-\bar{x}^2)^t = \begin{bmatrix} 99 \\ 97 \\ 106 \\ 97 \\ 0 \\ -300 \\ 0 \end{bmatrix} \otimes [99, 97, 106, 97, 0, -300, 0]$$

$$\bar{x}^2 = \begin{bmatrix} 0 & 2 & -7 & 2 & 99 & 399 & 99 \\ -2 & 0 & -9 & 0 & 97 & 397 & 97 \\ 7 & 9 & 0 & 9 & 106 & 406 & 106 \\ -2 & 0 & -9 & 0 & 97 & 397 & 97 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \\ -399 & -397 & -406 & -397 & -300 & 0 & -300 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \end{bmatrix}$$

$$\bar{x}^3 \otimes (-\bar{x}^3)^t = \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ 0 \\ 0 \\ -300 \end{bmatrix} \otimes [-110, 105, 164, 111, 0, 0, -300]$$

$$\bar{x}^3 = \begin{bmatrix} 0 & 5 & -54 & -1 & 110 & 110 & 410 \\ -5 & 0 & -59 & -6 & 105 & 105 & 405 \\ 54 & 59 & 0 & 53 & 164 & 164 & 464 \\ 1 & 6 & -53 & 0 & 111 & 111 & 411 \\ -110 & -105 & -164 & -111 & 0 & 0 & 300 \\ -110 & -105 & -164 & -111 & 0 & 0 & 300 \\ -410 & -405 & -464 & -411 & -300 & -300 & 0 \end{bmatrix}$$

Aplicando la operación Morfológica de Erosión obtenemos la Memoria MÍN

$$\Lambda = W = \bigwedge_{\mu=1}^p (\bar{x}^{\mu} \otimes (-\bar{x}^{\mu})^t)$$

Se comparan cada uno de los parámetros de las memorias para obtener el de menor valor.

$$W = \begin{bmatrix} 0 & 21 & 19 & 21 & 418 & 118 & 118 \\ -21 & 0 & -2 & 0 & 397 & 97 & 97 \\ -19 & 2 & 0 & 2 & 399 & 99 & 99 \\ -21 & 0 & -2 & 0 & 397 & 97 & 97 \\ -418 & -397 & -399 & -397 & 0 & -300 & -300 \\ -118 & -97 & -99 & -97 & 300 & 0 & 0 \\ -118 & -97 & -99 & -97 & 300 & 0 & 0 \end{bmatrix} \wedge \begin{bmatrix} 0 & 2 & -7 & 2 & 99 & 399 & 99 \\ -2 & 0 & -9 & 0 & 97 & 397 & 97 \\ 7 & 9 & 0 & 9 & 106 & 406 & 106 \\ -2 & 0 & -9 & 0 & 97 & 397 & 97 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \\ -399 & -397 & -406 & -397 & -300 & 0 & -300 \\ -99 & -97 & -106 & -97 & 0 & 300 & 0 \end{bmatrix} \wedge$$

$$\begin{bmatrix} 0 & 5 & -54 & -1 & 110 & 110 & 410 \\ -5 & 0 & -59 & -6 & 105 & 105 & 405 \\ 54 & 59 & 0 & 53 & 164 & 164 & 464 \\ 1 & 6 & -53 & 0 & 111 & 111 & 411 \\ -110 & -105 & -164 & -111 & 0 & 0 & 300 \\ -110 & -105 & -164 & -111 & 0 & 0 & 300 \\ -410 & -405 & -464 & -411 & -300 & -300 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 & -54 & -1 & 99 & 110 & 99 \\ -21 & 0 & -59 & -6 & 97 & 97 & 97 \\ -19 & 2 & 0 & 2 & 106 & 99 & 99 \\ -21 & 0 & -53 & 0 & 97 & 97 & 97 \\ -418 & -397 & -399 & -397 & 0 & -300 & -300 \\ -399 & -397 & -406 & -397 & -300 & 0 & -300 \\ -410 & -405 & -464 & -411 & -300 & -300 & 0 \end{bmatrix}$$

Después de obtener las memorias MÁX y MÍN ocupamos la función de Linear Associator Modificada.

Función Linear Associator Modificada:

$$LMAy = \begin{bmatrix} 99 & 98 & 98 \\ 111 & 111 & 111 \\ 119 & 120 & 121 \end{bmatrix}$$

FASE DE RECUPERACIÓN

Para la recuperación de "Y" patrón (palabra en inglés) se espera recibir un patrón de entrada "X" (palabra en español) con el cual se crearan dos vectores ruidosos A y B.

Para esto se usaran vectores que llamaremos U y \bar{U}

Estos vectores tomaran los siguientes valores.

$$U = \begin{pmatrix} 300 \\ 300 \\ 300 \end{pmatrix}$$

$$\bar{U} = \begin{pmatrix} -300 \\ -300 \\ -300 \end{pmatrix}$$

El primer patrón a recuperar será la palabra "vaca"

$$x^1 = \begin{bmatrix} 118 \\ 97 \\ 99 \\ 97 \end{bmatrix}$$

$$A = \begin{pmatrix} 118 \\ 97 \\ 99 \\ 97 \\ 300 \\ 300 \\ 300 \end{pmatrix}$$

$$B = \begin{pmatrix} 118 \\ 97 \\ 99 \\ 97 \\ -300 \\ -300 \\ -300 \end{pmatrix}$$

Ahora aplicamos le operación de dilatación al patrón A y a la memoria asociativa V para obtener un nuevo vector, y a este vector le extraeremos los tres últimos parámetros y los llamaremos C que se podría decir que es el vector U alterado por la operación de dilatación y el operador MÁX.

$$M\Delta A = \begin{bmatrix} 0 & 21 & 19 & 21 & 110 & 118 & 118 \\ -2 & 0 & -2 & 0 & 105 & 105 & 97 \\ 54 & 59 & 0 & 53 & 164 & 164 & 106 \\ 1 & 6 & -2 & 0 & 111 & 111 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ 201 & 203 & 194 & 203 & 0 & 0 & 300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix} \Delta \begin{bmatrix} 118 \\ 97 \\ 99 \\ 97 \\ 300 \\ 300 \\ 300 \end{bmatrix}$$

$$M\Delta A = \begin{bmatrix} 0 + 118 & 21 + 97 & 19 + 99 & 21 + 97 & 110 + 300 & 118 + 300 & 118 + 300 \\ -2 + 118 & 0 + 97 & -2 + 99 & 0 + 97 & 105 + 300 & 105 + 300 & 97 + 300 \\ 54 + 118 & 59 + 97 & 0 + 99 & 53 + 97 & 164 + 300 & 164 + 300 & 106 + 300 \\ 1 + 118 & 6 + 97 & -2 + 99 & 0 + 97 & 111 + 300 & 111 + 300 & 97 + 300 \\ 182 + 118 & 203 + 97 & 201 + 99 & 203 + 97 & 0 + 300 & 300 + 300 & 300 + 300 \\ 201 + 118 & 203 + 97 & 194 + 99 & 203 + 97 & 0 + 300 & 0 + 300 & 300 + 300 \\ 190 + 118 & 195 + 97 & 136 + 99 & 189 + 97 & 300 + 300 & 300 + 300 & 0 + 300 \end{bmatrix}$$

$$M\Delta A = \begin{bmatrix} 118\wedge & 118\wedge & 118\wedge & 118\wedge & 410\wedge & 418\wedge & 418 \\ 116\wedge & 97\wedge & 97\wedge & 97\wedge & 405\wedge & 405\wedge & 397 \\ 172\wedge & 156\wedge & 99\wedge & 150\wedge & 464\wedge & 464\wedge & 406 \\ 119\wedge & 103\wedge & 97\wedge & 97\wedge & 411\wedge & 411\wedge & 397 \\ 300\wedge & 300\wedge & 300\wedge & 300\wedge & 300\wedge & 600\wedge & 600 \\ 319\wedge & 300\wedge & 293\wedge & 300\wedge & 300\wedge & 300\wedge & 600 \\ 308\wedge & 292\wedge & 235\wedge & 286\wedge & 600\wedge & 600\wedge & 300 \end{bmatrix} = \begin{bmatrix} 118 \\ 97 \\ 99 \\ 97 \\ 300 \\ 293 \\ 235 \end{bmatrix}$$

Analizamos $C = \begin{bmatrix} 300 \\ 293 \\ 235 \end{bmatrix}$ todos los valores iguales a 300 se quedan así, y todos los valores

diferentes a 300 se pasan a 0 $C = \begin{bmatrix} 300 \\ 0 \\ 0 \end{bmatrix}$ si no es un vector One Hot Modificado entonces

pasamos a la siguiente etapa de lo contrario entonces hacemos el cálculo final sustituyendo todos los valores 300 por 1's.

Como paso final

$$LMAy \cdot C^t = \begin{bmatrix} 99 & 98 & 98 \\ 111 & 111 & 111 \\ 119 & 120 & 121 \end{bmatrix} \cdot [1 \ 0 \ 0] = \begin{bmatrix} 99 \\ 111 \\ 119 \end{bmatrix}$$

Patrón Obtenido

$$y = \begin{bmatrix} 99 \\ 111 \\ 119 \end{bmatrix} = \text{cow}$$

De igual manera se realizan los mismos pasos para recuperar las palabras faltantes.

El segundo patrón a recuperar será la palabra "caja"

$$x^2 = \begin{bmatrix} 99 \\ 97 \\ 106 \\ 97 \end{bmatrix}$$

$$A = \begin{pmatrix} 99 \\ 97 \\ 106 \\ 97 \\ 300 \\ 300 \\ 300 \end{pmatrix}$$

$$B = \begin{pmatrix} 99 \\ 97 \\ 106 \\ 97 \\ -300 \\ -300 \\ -300 \end{pmatrix}$$

$$M\Delta A = \begin{bmatrix} 0 & 21 & 19 & 21 & 110 & 118 & 118 \\ -2 & 0 & -2 & 0 & 105 & 105 & 97 \\ 54 & 59 & 0 & 53 & 164 & 164 & 106 \\ 1 & 6 & -2 & 0 & 111 & 111 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ 201 & 203 & 194 & 203 & 0 & 0 & 300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix} \Delta \begin{bmatrix} 99 \\ 97 \\ 106 \\ 97 \\ 300 \\ 300 \\ 300 \end{bmatrix}$$

$$M\Delta A = \begin{bmatrix} 99 \wedge & 118 \wedge & 125 \wedge & 118 \wedge & 410 \wedge & 418 \wedge & 418 \\ 97 \wedge & 97 \wedge & 104 \wedge & 97 \wedge & 405 \wedge & 405 \wedge & 397 \\ 153 \wedge & 156 \wedge & 106 \wedge & 150 \wedge & 464 \wedge & 464 \wedge & 406 \\ 100 \wedge & 103 \wedge & 104 \wedge & 97 \wedge & 411 \wedge & 411 \wedge & 397 \\ 281 \wedge & 300 \wedge & 307 \wedge & 300 \wedge & 300 \wedge & 600 \wedge & 600 \\ 300 \wedge & 600 \\ 289 \wedge & 292 \wedge & 242 \wedge & 286 \wedge & 600 \wedge & 600 \wedge & 300 \end{bmatrix} = \begin{bmatrix} 99 \\ 97 \\ 106 \\ 97 \\ 281 \\ 300 \\ 242 \end{bmatrix}$$

Analizamos $C = \begin{bmatrix} 281 \\ 300 \\ 242 \end{bmatrix}$ todos los valores iguales a 300 se quedan así, y todos los valores

diferentes a 300 se pasan a 0 $C = \begin{bmatrix} 0 \\ 300 \\ 0 \end{bmatrix}$ si es un vector One Hot Modificado entonces

transformamos todos los 300 en 1's y pasamos a la última etapa de recuperación

$$LMAy \cdot C^t = \begin{bmatrix} 99 & 98 & 98 \\ 111 & 111 & 111 \\ 119 & 120 & 121 \end{bmatrix} \cdot [0 \quad 1 \quad 0] = \begin{bmatrix} 98 \\ 111 \\ 120 \end{bmatrix}$$

Patrón Obtenido

$$y = \begin{bmatrix} 98 \\ 111 \\ 120 \end{bmatrix} = \text{box}$$

El tercer y último patrón a recuperar es la palabra **"niño"**

$$x^3 = \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \end{bmatrix}$$

$$A = \begin{pmatrix} 110 \\ 105 \\ 164 \\ 111 \\ 300 \\ 300 \\ 300 \end{pmatrix}$$

$$B = \begin{pmatrix} 110 \\ 105 \\ 164 \\ 111 \\ -300 \\ -300 \\ -300 \end{pmatrix}$$

$$M\Delta A = \begin{bmatrix} 0 & 21 & 19 & 21 & 110 & 118 & 118 \\ -2 & 0 & -2 & 0 & 105 & 105 & 97 \\ 54 & 59 & 0 & 53 & 164 & 164 & 106 \\ 1 & 6 & -2 & 0 & 111 & 111 & 97 \\ 182 & 203 & 201 & 203 & 0 & 300 & 300 \\ 201 & 203 & 194 & 203 & 0 & 0 & 300 \\ 190 & 195 & 136 & 189 & 300 & 300 & 0 \end{bmatrix} \Delta \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ 300 \\ 300 \\ 300 \end{bmatrix}$$

$$M\Delta A = \begin{bmatrix} 110\Lambda & 126\Lambda & 183\Lambda & 132\Lambda & 410\Lambda & 418\Lambda & 418 \\ 108\Lambda & 105\Lambda & 162\Lambda & 111\Lambda & 405\Lambda & 405\Lambda & 397 \\ 164\Lambda & 164\Lambda & 164\Lambda & 164\Lambda & 464\Lambda & 464\Lambda & 406 \\ 111\Lambda & 111\Lambda & 162\Lambda & 111\Lambda & 411\Lambda & 411\Lambda & 397 \\ 292\Lambda & 308\Lambda & 365\Lambda & 304\Lambda & 300\Lambda & 600\Lambda & 600 \\ 311\Lambda & 308\Lambda & 358\Lambda & 304\Lambda & 300\Lambda & 300\Lambda & 600 \\ 300\Lambda & 300\Lambda & 300\Lambda & 300\Lambda & 600\Lambda & 600\Lambda & 300 \end{bmatrix} = \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ 292 \\ 300 \\ 300 \end{bmatrix}$$

Analizamos $C = \begin{bmatrix} 292 \\ 300 \\ 300 \end{bmatrix}$ todos los valores iguales a 300 se quedan así, y todos los valores

diferentes a 300 se pasan a 0 $C = \begin{bmatrix} 0 \\ 300 \\ 300 \end{bmatrix}$ en esta ocasión no es un vector One Hot Modificado

entonces hacemos el cálculo de siguiente:

$$W\nabla B = \begin{bmatrix} 0 & 2 & -54 & -1 & 99 & 110 & 99 \\ -21 & 0 & -59 & -6 & 97 & 97 & 97 \\ -19 & 2 & 0 & 2 & 106 & 99 & 99 \\ -21 & 0 & -53 & 0 & 97 & 97 & 97 \\ -418 & -397 & -399 & -397 & 0 & -300 & -300 \\ -399 & -397 & -406 & -397 & -300 & 0 & -300 \\ -410 & -405 & -464 & -411 & -300 & -300 & 0 \end{bmatrix} \nabla \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ -300 \\ -300 \\ -300 \end{bmatrix}$$

$$W\nabla B = \begin{bmatrix} 110 \text{ V} & 107 \text{ V} & 110 \text{ V} & 110 \text{ V} & -201\text{V} & -190\text{V} & -201 \\ 89 \text{ V} & 105 \text{ V} & 105 \text{ V} & 105 \text{ V} & -203\text{V} & -203\text{V} & -203 \\ 91 \text{ V} & 107 \text{ V} & 164 \text{ V} & 113 \text{ V} & -194\text{V} & -201\text{V} & -201 \\ 89 \text{ V} & 105 \text{ V} & 111 \text{ V} & 111 \text{ V} & -203\text{V} & -203\text{V} & -203 \\ -308\text{V} & -292\text{V} & -235\text{V} & -286\text{V} & -300\text{V} & -600\text{V} & -600 \\ -289\text{V} & -292\text{V} & -242\text{V} & -286\text{V} & -600\text{V} & -300\text{V} & -600 \\ -300\text{V} & -300\text{V} & -300\text{V} & -300\text{V} & -600\text{V} & -600\text{V} & -300 \end{bmatrix} = \begin{bmatrix} 110 \\ 105 \\ 164 \\ 111 \\ -235 \\ -242 \\ -300 \end{bmatrix}$$

- Ahora nos enfocamos en el vector $D = \begin{bmatrix} -235 \\ -242 \\ -300 \end{bmatrix}$, todos los valores iguales a -300 se

mantienen y los que sean distintos se pasan 0, entonces si el vector $D = \begin{bmatrix} 0 \\ 0 \\ -300 \end{bmatrix}$ tiene la

forma de un Zero Hot modificado entonces cambiamos todos los valores -300 a 1's y realizamos el último paso de nuestro algoritmo.

$$LMAy \cdot D^t = \begin{bmatrix} 99 & 98 & 98 \\ 111 & 111 & 111 \\ 119 & 120 & 121 \end{bmatrix} \cdot [0 \ 0 \ 1] = \begin{bmatrix} 98 \\ 111 \\ 121 \end{bmatrix}$$

Patrón Obtenido

$$y = \begin{bmatrix} 98 \\ 111 \\ 121 \end{bmatrix} = boy$$

SIMBOLOGÍA

- M** Memoria asociativa morfológica *max.*
- W** Memoria asociativa morfológica *min*
- m_{ij}** *ij-ésima* componente de la memoria asociativa **M**
- B** Conjunto al que pertenecen las componentes m_{ij}
- \in** Pertenecía de un elemento aun conjunto
- x** Vector columna que corresponde a un patrón de entrada
- y** Vector columna que corresponde a un patrón de salida
- A** Conjunto al que pertenece los componentes x y y .
- (x, y)** Asociación de un patrón de entrada con uno de salida
- (x^k, y^k)** Asociación de la *k-ésima* pareja de patrones
- $(x^\mu)^t$** Traspuesto del vector x^μ
- n** Dimensión de los patrones de entrada
- m** Dimensión de los patrones de salida
- p** Número de parejas del conjunto fundamental
- \vee** Operador máximo
- \wedge** Operador mínimo
- \cdot** Productor usual entre vectores o matrices
- ∇** Producto máximo (entre matrices)
- Δ** Producto mínimo
- \forall** Cuantificador universal
- \exists** Cuantificador existencial
- \otimes** Operador N-ario