UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

Uso de la minería de texto para la clasificación automática de preguntas abiertas en el contexto de la enseñanza de las ciencias experimentales

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Licenciado en Ciencias de la Computación

PRESENTA:

Rafael Robles Rios

TUTOR

Gustavo De la Cruz Martínez.

Ciudad Universitaria, CD. MX, 2021.

A todos mis amigos, profesores y futuros colegas que me ayudaron en la elaboración de esta tesis, infinitas gracias por toda su ayuda y buena voluntad.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Se describe a la lingüística computacional como un campo que se apoya en la lingüística y la computación, pero pone especial atención a los aspectos lingüísticos del lenguaje humano. Su objetivo es la construcción de modelos computacionales que representan uno o más aspectos de este lenguaje, y así desarrollar software que permita realizar el análisis automático de aspectos como la fonética, la fonología, la morfología, la sintaxis, la semántica y la pragmática. Esto ha apoyado diferentes actividades como la enseñanza de idiomas extranjeros, corrección ortográfica y sintáctica de textos, reconocimiento de la voz humana y su procesamiento para identificar la información contenida en las frases pronunciadas, entre muchas otras.

El área que se enfoca en el procesamiento de representaciones del lenguaje humano en texto, se conoce como minería de texto. Las técnicas de minería de texto están enfocadas a la exploración de conocimiento dentro de uno o varios conjuntos de texto. Con estas técnicas es posible distinguir categorías gramaticales, realizar análisis de sentimiento, determinar el tema o conceptos usados en el texto, el agrupamiento de varios textos según su contenido, entre otras tareas.

Este trabajo se centra en la identificación de conceptos, temas o ideas, los cuales una vez son identificados, es posible determinar los textos que traten del mismo concepto, incluso cuando no utilicen la misma terminología. Una estrategia utilizada en la minería de textos para determinar los conceptos, temas o ideas principales tratadas en los documentos, se basa en extraer el conjunto de términos representativos del contenido de los textos. Así como, la capacidad para encontrar un concepto que se basa en el análisis de las ocurrencias de determinados términos y combinaciones de los términos en el documento.

1.1. Objetivo

El objetivo de esta tesis es plantear una metodología para hacer la clasificación automática de textos, para apoyar el análisis de la comprensión de los estudiantes del bachillerato en temas de física y biología, dentro del contexto de los nuevos laboratorios de ciencias del bachillerato de la UNAM.

La metodología planteada se utilizará para analizar las respuestas de los estudiantes a preguntas abiertas utilizando técnicas de minería de texto, generando modelos que ayuden a los investigadores al análisis de los conocimientos científicos de los estudiantes del bachillerato.

1.2. Hipótesis

Es posible evaluar un cuestionario de preguntas abiertas de forma parcial, para entrenar a un clasificador automático que logre concluir la evaluación siguiendo los criterios de los investigadores; logrando el mismo resultado, que si la encuesta fuera evaluada de forma total por los investigadores.

Para comprobar esto, se comparan los conceptos y sus relaciones en cada clase identificada, por cada una de las preguntas, si estas relaciones son semejantes entre las dos evaluaciones, se puede concluir que la evaluación automática será equivalente a la evaluación manual.

1.3. Descripción general

Como se ha señalado, el uso de la minería de textos para la identificación de conceptos, se ha utilizado para el análisis de textos científicos con el fin de identificar la temática principal. También se utiliza para el análisis de encuestas y cuestionarios de preguntas abiertas, lo que permite identificar rápidamente tendencias en las respuestas de los entrevistados. Las preguntas abiertas se pueden utilizar para obtener la opinión de una persona sobre un determinado tema, o bien conocer el nivel de comprensión de un tema dado y su habilidad de expresar el conocimiento del mismo. En el contexto de la enseñanza de las ciencias experimentales, el análisis de las respuestas de este tipo de preguntas ayuda a determinar el nivel de comprensión de los estudiantes.

El análisis tradicional de estos cuestionarios es manual, es decir, los investigadores analizan una a una las respuestas de los estudiantes, por lo que esta tarea puede llevar bastante tiempo. Y es necesario tener la participación de un grupo de expertos para evaluar correctamente las respuestas, ya que ellos cuentan con un amplio conocimiento sobre el tema que se está evaluando, y en ocasiones puede generar conflictos de opiniones entre ellos, como:

- La introducción de un sesgo personal.
- Despreciar conceptos en una etapa temprana del análisis.
- Tomar un enfoque distinto al objetivo general.

En el caso de las preguntas abiertas se busca hacer una clasificación automática sobre el nivel de competencia, usando un entrenamiento supervisado, es decir, entrenar a un modelo con una muestra previamente calificada para poder clasificar cada pregunta.

En esta tesis se presentará el proceso de evaluación asistida, el cual emplea las técnicas de minería de textos para apoyar a los investigadores en el proceso de asignación del nivel de competencia en un tiempo menor, ya que como se mencionó anteriormente, este análisis es manual y puede tomar bastante tiempo. Para ello se construirá un modelo capaz de clasificar o asignar automáticamente un nivel de competencia a las respuestas de los estudiantes. Después se realizará una comparación entre la muestra evaluada por los investigadores y la muestra evaluada por el modelo, para comprobar la eficiencia del modelo, es decir, comprobar que el modelo será capaz de dar una asignación similar a la de los investigadores.

Este modelo será entrenado con una muestra del cuestionario elaborado para el proyecto de investigación: “Procesos de transformación de las representaciones científicas en los estudiantes del bachillerato bajo un entorno multi representacional apoyado con

tecnologías digitales", centrado en la investigación de la enseñanza de las ciencias en estudiantes de bachillerato.

Este cuestionario cuenta con 8 preguntas donde cada una consta de 387 respuestas, el cuestionario fue evaluado con un nivel de competencia para cada una de las respuestas, el cual establece el dominio que tienen los estudiantes sobre el tema de genética.

Los niveles de competencia están definidos en la rúbrica creada por los investigadores [1]. El entrenamiento se realizará usando los algoritmos de minería, los cuales implementan la búsqueda de nuevos patrones o reglas para realizar una clasificación similar a la que tiene la muestra de entrenamiento.

El modelo será personalizado en las etapas generales de la minería de texto para que se ajuste a las necesidades del investigador; esta personalización se realizará por medio de una aplicación web que permitirá realizar varios experimentos de una forma más sencilla y práctica, donde las respuestas serán cargadas en una base de datos para que las tareas de preproceso de la información sean aplicadas automáticamente. Además, la aplicación web nos permitirá entrenar varias veces al modelo, con el fin de obtener el mejor modelo posible y usarlo para evaluar respuestas que aún no cuenten con un nivel de competencia. Por último, se comparará la asignación realizada por el modelo y la asignación de los investigadores usando las tablas y las gráficas que generará dicha aplicación.

El contenido de esta tesis está organizado de la siguiente forma:

- En el capítulo 2, se comenzará a hablar de la minería de datos y la transformación que tienen los datos en cada una de sus etapas. Después, se ampliará esta visión usando datos textuales donde se mencionan las etapas generales de la minería de textos, las cuales son: preprocesamiento, aplicación de los algoritmos de minería de texto, visualización de la información y refinamiento de los resultados. También se mencionara el funcionamiento teórico de los algoritmos de minería de textos, los cuales serán los árboles de decisión, los clasificadores probabilísticos, las redes neuronales y las máquinas de soporte vectorial.

- En el capítulo 3, se hablará de la investigación de la enseñanza de las ciencias y el uso de cuestionarios como instrumento para analizar las representaciones externas de los estudiantes, donde tales representaciones contienen elementos cognitivos que reflejan los procesos y concepciones de los estudiantes. Después, se mencionara la propuesta del proceso de evaluación asistida y la aplicación web que será creada como parte de este trabajo, la cual contará con funciones básicas para las actividades de los investigadores, como la carga de archivos para el entrenamiento y la creación de un modelo para la evaluación automática, así como un apartado para visualizar el contenido de las respuestas y su nivel de competencia.

- En el capítulo 4, se hablará de las pruebas de desempeño realizadas con los diferentes algoritmos de minería de textos, para determinar cuál de ellos es el mejor para este análisis e implementarlo en la aplicación. Adicionalmente, se mostrarán los resultados que se obtengan con el modelo entrenado y los resultados de la comparación entre ambas evaluaciones.

- En el capítulo 5, se discutirá las conclusiones a las que se llegaron al usar la propuesta que realiza el proceso de evaluación asistida, y cómo puede ayudar a los investigadores a implementar un análisis más sencillo y rápido. También se mencionan algunas sugerencias para próximos estudios.

Capítulo 2

Minería de datos

La minería de datos se define como: el proceso de descubrimiento de patrones útiles y potencialmente novedosos que hay en grandes cantidades de datos, también conocidas como, fuentes de datos [2]. Algunas fuentes de datos pueden incluir bases de datos, almacenes de datos, información de la web u otro tipo de información estructurada. Un ejemplo del uso de la minería de datos son los sistemas de recomendación automática de música, los cuales analizan la información de varios usuarios y descubren patrones entre sus preferencias, para recomendar música basándose en dichas preferencias.

2.1. Proceso de descubrimiento de información

En [2] mencionan que la minería de datos es una etapa del proceso de descubrimiento de conocimiento (por sus siglas en inglés KDD o Knowledge Discovery from Data). El proceso de descubrimiento de información también es llamado como "minería de datos a partir de los datos" o simplemente "minería de datos". Este último es más utilizado por su brevedad. Las etapas de minería de datos están basadas en el tratamiento que necesitan los datos para descubrir información implícita que hay en ellos, las etapas que usa el proceso KDD son: limpieza de datos, integración de datos, selección de datos, transformación de datos, minería de datos, evaluación y análisis, finalmente la representación del conocimiento. En la figura 2.1 se muestra el flujo que sigue el proceso KDD.

2.1.1. Limpieza de datos

Las fuentes de datos que se analizan pueden o no contener irregularidades en los datos, como datos incompletos, inconsistentes o redundantes, si los datos se encuentran con irregularidades son poco confiables, así como el resultado del proceso de minería de datos. Para solucionar este problema, se aplica la etapa de limpieza de datos, la cual busca eliminar las irregularidades en las fuentes de datos. La etapa de limpieza de datos trabaja completando los valores que hacen falta, identificando o eliminando los valores atípicos y resolviendo las inconsistencias. Por ejemplo: en una base de datos para un sistema de recomendación de música, puede contener un registro con el nombre de la canción y no tener asignado un género musical, lo cual haría imposible recomendar la canción, si el recomendador utiliza el género musical para encontrar nueva canciones.

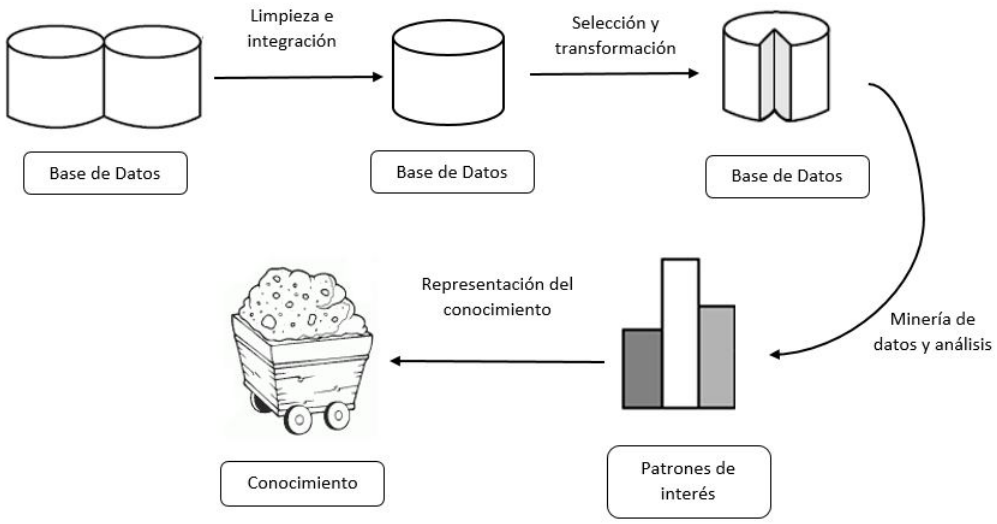


Figura 2.1: Muestra las etapas del modelo KDD.

Entonces en esta etapa es necesario eliminar este registro o bien asignar un género musical a ese registro.

2.1.2. Integración de datos

En esta etapa se trabaja con varias fuentes de datos y su fin es convertirlas en una sola fuente de datos, los problemas que puede ocurrir al unir dos o más fuentes de datos son: inconsistencias y redundancias.

Las inconsistencias pueden pasar cuando un elemento está en dos o más fuentes de datos con diferentes nombres, es decir, este elemento representa la misma información, solo que tiene dos registros distintos, lo que causaría confusión en el análisis.

La redundancia ocurre cuando se tienen almacenadas varias copias del mismo dato en una sola fuente de datos, si esto sucede provocaría un mal análisis. En el ejemplo del recomendador de música, se puede integrar la información sobre la actividad en las redes sociales de los usuarios, para conocer las aplicaciones de música que usan o los grupos musicales que son de su interés y así tener más información para hacer una recomendación.

2.1.3. Selección de datos

Siempre es necesario recordar la finalidad del análisis, porque en esta parte se toman sólo los datos que sean relevantes para dicho análisis. La selección de datos debe mantener la integridad y la representación de los datos, porque la finalidad es crear una versión reducida de los datos actuales, con la menor cantidad de datos posibles.

En el ejemplo del sistema de recomendación de música, solo se necesita analizar la preferencia y frecuencia de la música que han escuchado los usuarios, omitiendo

algunos de los datos personales como el nombre de los usuarios y su dirección.

2.1.4. Transformación de datos

En esta etapa los datos se cambian de formato para que en la siguiente etapa sea más sencillo encontrar e interpretar los patrones que hay dentro de los datos, para lograr esto, tradicionalmente se aplica la agregación de datos resumidos, normalización y discretización [2].

En la agregación de datos resumidos, se aplican operaciones de resumen sobre los datos actuales, un ejemplo de esto es considerando las ventas diarias de algún producto, los datos de las ventas se pueden organizar y guardar de forma semanal, mensual y anual, con esto se crea tres representaciones de los datos y así es posible detectar ciertos patrones. Con la normalización se escalan los valores de los datos dentro de un rango definido, casi siempre se usa el rango de 0 a 1, aunque esta escala puede cambiar en función del análisis.

La discretización reemplaza los valores originales de los datos por un rango o etiquetas conceptuales que representan a los datos. Generalmente se aplica el reemplazo sobre valores numéricos, por ejemplo las edades pueden ser sustituidas por los rangos: 0 a 10, 11 a 20, 21 a 60. Con el reemplazo de etiquetas conceptuales se elige una palabra que represente al valor que se va a sustituir, por ejemplo con las edades se puede reemplazar por las siguientes, las edades de 0 a 10, con la etiqueta conceptual “infante”, las edades de 11 a 21 por “juvenil” y de 21 a 60 como “adulto”. Usando las etiquetas conceptuales podemos crear varias jerarquías para diferenciar a los datos.

2.1.5. Algoritmos de Minería de datos

En esta etapa se aplican los algoritmos de minería de datos para obtener la información oculta que hay en los datos. Ya que dependiendo de la información obtenida de las etapas anteriores, el algoritmo buscará patrones que puedan dar a conocer nueva información o confirmar hipótesis sugeridas en el análisis, por esto es la etapa de mayor importancia del proceso.

Funcionalidad de los algoritmos de minería de datos

Los algoritmos de minería realizan las siguientes funciones: extracción de patrones frecuentes, asociaciones y correlaciones, clasificación, regresión, análisis de agrupamientos, análisis de valores atípicos y vizualización.

Los algoritmos de minería con funcionalidad predictiva, realizan el descubrimiento de patrones sobre los datos actuales buscando asociaciones y correlaciones que hay en los datos, posteriormente realizan predicciones sobre un conjunto de datos usando estos patrones. Los algoritmos de minería con funcionalidad descriptiva agrupan a los datos en subconjuntos observando a las características que tienen en común. De esta manera se especifica la tarea que realiza el algoritmo de minería para encontrar los patrones que hay dentro de los datos.

2.1.6. Evolución y análisis

Los algoritmos de minería de datos tiene el potencial de generar cientos o miles de patrones o reglas, pero solo algunas de estas reglas son útiles para obtener información relevante.

Los patrones con información relevante se distinguen por ser de interés para el desarrollo de una investigación o para validar nueva información, también pueden validar una hipótesis planteada previamente.

Existen varias medidas para determinar un patrón relevante, estas medidas se basan en su estructura y las estadísticas subyacentes. Una de ellas es medir el número de transacciones con la base de datos que son necesarias para cumplir la regla, otra de ellas es evaluar el grado de confianza en el patrón para el uso de nueva información. También existen medidas para determinar patrones relevantes que incluyen la precisión y las reglas de clasificación, en términos generales, la precisión nos dice el porcentaje de datos que un patrón o regla predice correctamente [2].

En general, cada medida está asociada con un umbral, que puede puede ser controlado en el análisis, por ejemplo podemos considerar a los patrones que tengan un umbral de precisión mayor al 70 % como patrones relevantes, ya que es probable que los que estén por debajo de este umbral presenten ruido, excepciones o casos frontera y sean patrones no relevantes.

También existen medidas subjetivas para medir la utilidad de un patrón, estas medidas se basan en las creencias de los investigadores en los datos. Estas medidas encuentran patrones interesantes si los patrones son inesperados, contradiciendo la creencia que se tenía previamente u ofrecen nueva información. Pero también los patrones que se esperan pueden ser interesantes, si confirman una hipótesis que se desea validar o si se parece a un presentimiento que se tenía sobre los datos.

Estas medidas son esenciales para el descubrimiento eficiente de patrones relevantes. Dichas medidas se pueden usar después de aplicar el algoritmo de minería de datos para clasificar los patrones descubiertos y obtener solo los patrones de interés. Tales medidas pueden usarse para guiar y restringir el proceso del descubrimiento, mejorando la eficacia de la búsqueda al eliminar patrones que no satisfacen las restricciones.

2.1.7. Representación del conocimiento

Una vez obtenida la información se debe elegir una forma para mostrarla, esta información puede ser representada por medio de tablas, gráficas, grafos dirigidos, imágenes de píxeles y proyecciones de figuras geométricas [2] o bien una implementación específica como una interfaz gráfica.

2.2. Minería de textos

La minería de texto se puede considerar como un proceso análogo al proceso de minería de datos, son análogos porque comparten la misma finalidad y algoritmos de minería, ambos buscan descubrir nueva información en los datos. La diferencia radica en el tipo de fuente de datos que utilizan. La minería de datos utiliza una fuente de datos que puede ser numérica o textual, como en la minería de texto, la cual puede trabaja con datos textuales. Aunque estos datos pueden ajustarse según el análisis. Se define a la minería de texto como: el proceso de descubrimiento de información y conocimiento que no se conocían, a partir de datos textuales [3]. De forma general, la minería de textos es un proceso que utiliza los datos textuales para transformarlos y crear un modelo que refleja el conocimiento obtenido de los datos textuales, finalmente usando el conocimiento para obtener conclusiones de los datos textuales.

2.2.1. Aplicaciones de minería de texto

Las aplicaciones de la minería de textos están enfocadas en analizar grandes cantidades de datos textuales para obtener conocimiento puntual de esos datos. Algunas de sus aplicaciones mencionadas en [3] son:

- **Identificación de hechos y datos puntuales**: consiste en identificar y extraer las referencias a los objetos que hay en los datos textuales como: personas, instituciones, eventos y la relaciones entre ellos.

  En el contexto de nuestro ejemplo de un recomendador de música automática, se puede analizar las notas de periódico que hablen sobre conciertos de música, para obtener relaciones entre los objetos (grupos, asistentes, etc.) que ahí se mencionan, por ejemplo:

  - ”La banda mexicana JotDog encendió el ánimo de sus seguidores, en donde derrochó talento con su sonido pop, con tintes electrónicos”.

  De la nota se extraen las referencias: JotDog, pop, electrónico; además, encontramos las relaciones entre el grupo JotDog y los géneros de música pop y electrónica.

- **Agrupamiento de textos similares:** se agrupan varios textos por su similitud, para determinar la similitud entre los textos se puede considerar la terminología usada en ellos, aunque cada investigador puede crear su propia definición de similitud.

  Continuando con el ejemplo del recomendador de música, podemos agrupar los comentarios por los artistas mencionados en estos, por ejemplo:

  - “Gracias a **Los Caligaris** por haberle regalado mucha felicidad, al público”.
  - “**Gorillaz**. Felicidades Vive, te mereces un aplauso”.

  Cada comentario sería asignado a un grupo, Los Caligaris y Gorillaz, respectivamente.

- **Clasificación de textos:** asigna clases o etiquetas a textos de forma automática. Con esta estrategia se puede determinar el tema que es tratado en el documento.

  Como ejemplo podemos clasificar los comentarios de redes sociales sobre el próximo lanzamiento de un nuevo disco de un artista, la clasificación sería sobre el comentario identificando si lo comprará o no. Así obteniendo una visión general de las posibles ventas del disco.

- **Análisis de sentimiento:** se trata de determinar el sentimiento de un comentario, para entender la opinión o postura sobre un hecho. Esta opinión puede ser positiva si está de acuerdo con que se describe del hecho o negativa si no está de acuerdo o neutra si no le importa lo que se describe.

  En nuestro ejemplo del recomendador de música, nos interesa saber la opinión que tiene los usuarios sobre los artistas, esto se puede obtener analizando el sentimiento de los comentarios de los usuarios, por ejemplo en el comentario:

  - “Gracias Juanes y Mon Laferte por tan bellos momentos”

  En este comentario se observa un sentimiento positivo.

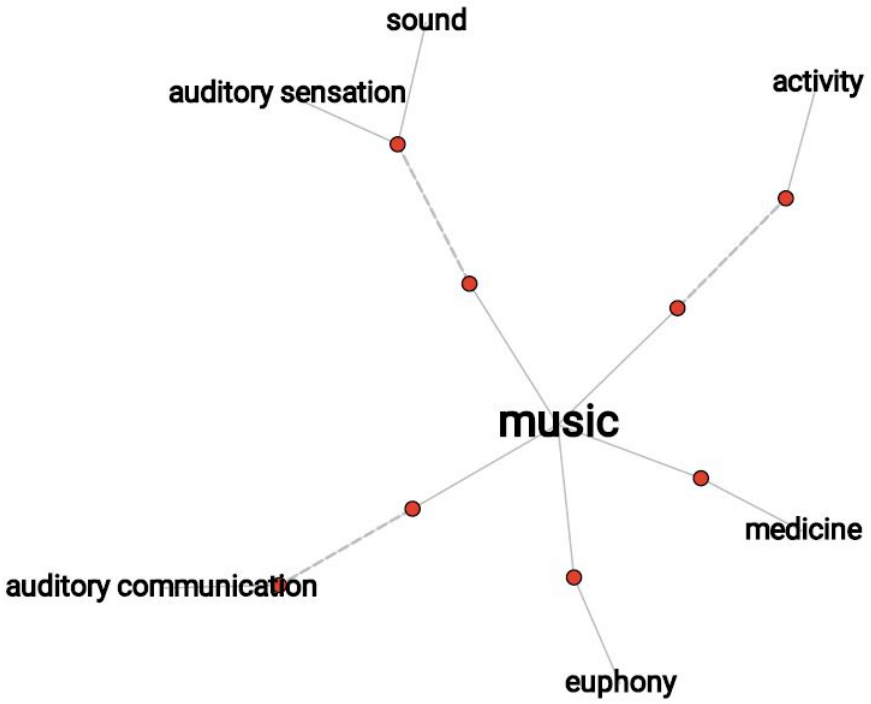


Figura 2.2: Muestra las relaciones que tiene el tesauro sobre la palabra ”music”.

- **Visualización de información:** tiene como objetivo mostrar la información de manera que sea fácil de navegar entre los datos textuales y los resultados obtenidos, para analizar los resultados de una manera sencilla. Por ejemplo, un visual thesaurus, es una herramienta que contiene un índice relacionado entre temas similares. Si buscamos la palabra “music” aparece una red de conceptos, la cual nos permite navegar entre los conceptos que están relacionados, a continuación se muestran en la figura 2.2 las palabras relacionadas como; sound, medicine y activity.

2.3. Componentes del proceso de minería de texto

En [4], se menciona que el proceso de minería de texto consta de cuatro etapas, las cuales son: tareas de preprocesamiento, aplicación de los algoritmos de minería, presentación o visualización de los resultados y refinamiento del proceso.

- **Tareas de preprocesamiento:** estas tareas son para ajustar los datos textuales a un formato específico. Algunas de estas tareas son: integrar varias fuentes de datos textuales, eliminación de palabras irrelevantes, pasar todo el texto a un formato en especifico, o transformar a un modelo de representación de documentos.
- **Aplicación de algoritmos de minería:** en esta etapa se aplican los algoritmos de minería de texto, obteniendo los patrones en la información de los datos textuales.

Procesos de minería de datos	Proceso de minería de textos
Limpieza de datos, Integración de datos, Selección de datos, Transformación de datos	Tareas de preprocesamiento
Minería de datos	Aplicación de los algoritmos de minería
Representación de la información	Visualización de los resultados
Evaluación y análisis	Refinamiento del proceso

Cuadro 2.1: Muestra las etapas equivalentes entre los procesos de minería de datos y minería de textos.

- **Visualización de la información:** se muestra la información obtenida de una manera sencilla de observar y navegar a través de ella, con la finalidad de analizar y evaluar la utilidad de la información, siempre teniendo en cuenta el objetivo a resolver. Las formas más comunes de visualización de la información son las gráficas o tablas.
- **Refinamiento:** a partir de los resultados obtenidos de los algoritmos de minería, se puede considerar repetir algunas etapas del proceso de minería de textos, con algún cambio en la ejecución del proceso. Por mencionar un cambio, puede ser probar con otro algoritmo de minería, o bien, hacer ajustes en el preprocesamiento como la eliminación de palabras irrelevantes. Este refinamiento tiene como objetivo encontrar un resultado más adecuado al objetivo a resolver.

De acuerdo con la descripción anterior, las etapas descritas en el proceso de minería de datos tiene una equivalencia con las etapas del proceso de minería de textos, mostrada en el cuadro 2.1, esta equivalencia se basa en el tratamiento y uso de los datos en cada una de las etapas de los respectivos procesos.

A continuación se detallan las tareas de preprocesamiento y los algoritmos de clasificación de minería para el uso de texto, ya que es una parte importante de este trabajo.

2.4. Tareas de preprocesamiento

Las tareas de preprocesamiento son procesos que se aplican en los textos para analizar la información que se encuentra dentro de los documentos, estos procesos pueden eliminar palabras (*stopwords*), sustituir términos que expresan un mismo concepto, entre otros. Esto se hace para crear una selección de la información que pueda ser más significativa al análisis, en nuestro caso seleccionar la información para el aprendizaje del algoritmo de minería. Como primer paso se puede comenzar eliminando saltos de línea y los caracteres que no sean necesarios para el análisis, por ejemplo; exclamaciones, signos de interrogación, entre otros.

2.4.1. Stopwords

Usualmente la lista de stopwords son las palabras que tienen mayor ocurrencia en los documentos y no aportan información útil al análisis, así que pueden ser eliminadas en esta etapa. La lista de stopwords más usuales están disponibles en la red. Las palabras más usadas como candidatas a stopwords son: artículos, preposiciones y conjunciones, con estas palabras se realiza una búsqueda en los documentos para eliminarlas

y así se obtiene una reducción de la cantidad de términos usados en los documentos, manteniendo la información relevante que estos tienen.

2.4.2. Stemming

Las fuentes de datos textuales pueden contener varios documentos con muchos términos distintos que expresan el mismo concepto, esto es propio del lenguaje natural. Normalmente cuando se analizan las frecuencias de los términos se considera toda la cadena de caracteres, es decir, que cada uno de los términos son distintos en lugar de agrupar los que son similares, lo que puede causar que se pierda información relevante. Este error puede ser solucionado con algún método de normalización. El stemming puede ser uno de ellos, ya que propone unir bajo un único término a las palabras que tiene un origen común, a este único término es conocido como: stem o raíz.

También existe la lingüística estructural, la cual define la palabra raíz por su posición gramatical en el texto. Así mismo, se pueden usar los n-gramas que son una serie de palabras consecutivas dentro de una oración. El uso de la lingüística estructural y los n-gramas puede ser más complicada, por este motivo se recomienda usar técnicas más sencillas como el stemming. El stemming usa algoritmos que tratan de eliminar a los sufijos de las palabras, obteniendo el mínimo de caracteres posibles con el máximo de información de los documentos. Lo que permite relacionar un término con otros u otros de su misma forma.

Los algoritmos que usa el stemming pueden encontrar las raíces empleando las variantes de género y número, incluso hay algoritmos que combinan esto con una lista de sufijos y prefijos. Por ejemplo, si tenemos que distinguir entre “bibliotecario”, “bibliotecaria”, “bibliotecarios” y “bibliotecarias”, todos los términos anteriores serían reducidos por el stemming a su palabra raíz que es “biblioteca”. Existen errores derivados de cortar incorrectamente las palabras, es decir, cuando cortamos un sufijo demasiado corto y no se recupera todo lo esperado, lo que origina incongruencia en los textos. También puede ocurrir que se eliminé una parte demasiado larga, es decir, dejamos una raíz muy corta, la cual coincidirá con varias palabras causando confusión.

Otro problema es la pérdida de parte del sentido, por ejemplo si tenemos “gravitatorio”, que se refiere a la fuerza de la gravedad, lo reducimos a “grave”, con el sentido de serio, estamos asociando un término a conceptos distintos.

Hay que señalar que el stemming no solo se aplica a palabras que usan variantes de la misma, sino que también se puede aplicar para sustituir palabras que son distintas, pero donde ambas representan el mismo concepto, como “gato” y “minino”. Cuando se aplica la tarea de stemming hay que tener en cuenta, el lenguaje que se emplea en los documentos, ya que cada lenguaje cuenta con excepciones y algunas ambigüedades. La longitud del o de los documentos. De forma general, el stemming se puede ver como una sustitución de términos por su respectiva raíz, de este modo aumenta el porcentaje de documentos que contienen el mismo término o raíz.

El algoritmo de Porter

El algoritmo de Porter elimina los sufijos basándose en dos reglas de eliminación, las cuales según [5] son:

- El sufijo que se elimina siempre es el más largo.
- Al eliminar el sufijo, se mantiene una longitud mínima.

Palabra	cada	padre	da	la	mitad	de	genética	por
Frecuencia	1	1	1	1	1	1	1	1

Palabra	eso	el	bebe	tiene	rasgos	ambos	información
Frecuencia	1	1	1	1	1	1	1

Cuadro 2.2: Muestra la representación de un documento a un vector de característica.

El algoritmo de porter tiene una lista de sufijos, la cual ayuda a la identificación del sufijo a eliminar, así mismo la forma para determinar la longitud de la palabra está dada por la fórmula 2.1, que representa una sucesión de vocales (representado por letra V) y consonantes (representado por letra C).

$$[C](VC)^n[V] \tag{2.1}$$

Teniendo en cuenta estas dos reglas, es posible aplicar varias veces el algoritmo de Porter a cada una de las palabras que hay en los documentos, con la finalidad de eliminar todos los sufijos posibles. Por ejemplo, si tenemos la palabra "presentaciones", el algoritmo buscará el afijo más largo que es "ciones", y al eliminar este sufijo se mantiene la longitud adecuada para mantener el contexto, obteniendo como resultado la palabra "presenta".

2.4.3. Representación de documentos

Los algoritmos de minería de texto no procesan directamente los documentos, estos documentos son convertidos en una representación más manejable. Comúnmente se transforman en vectores de características o vectores de palabras. Una característica es una entidad de un conjunto, en nuestro caso es un término o palabra. Un documento puede ser representado por su vector de palabras como una secuencia de palabras y sus frecuencias. Por ejemplo, si tenemos el siguiente texto: "Cada padre da la mitad de la información genética por eso el bebe tiene rasgos de ambos", su vector de palabras está reflejado en el cuadro 2.2.

Otro método para representar documentos es la bolsa de palabras, este es el más sencillo. Simplemente usa todas las palabras de un documento como características, por lo tanto la dimensión del espacio de la bolsa de palabras es igual al número de palabras diferentes en todo el documento.

También existen métodos que utilizan un peso en lugar de la frecuencia, este peso es calculado para cada palabra. El más simple es el binario, donde su peso es uno si la palabra correspondiente está en el documento o cero en caso contrario.

Un método de ponderación más complejo es el esquema TF-IDF, el cual toma en cuenta las frecuencias de las palabras en el documento y su categoría. El esquema TF-IDF calcula el peso de la palabra w del documento d, en la fórmula 2.2 se muestra como se calcula el peso, donde $TermFreq(w,d)$ es la frecuencia de la palabra w en el documento d, N es el número de todos los documentos y $Docfreq(w)$ es el número de documentos que cuentan con la palabra w [4]:

$$\text{TF-IDF}(w,d) = TermFreq(w,d) * log(N/Docfreq(w)) \tag{2.2}$$

2.5. Algoritmos de clasificación en la minería de texto

Dentro de los algoritmos de minería de tipo predictivo, usaremos a los clasificadores, los cuales son capaces de decidir a qué categoría o clase pertenece un elemento. Los clasificadores de texto toman un documento de la fuente de datos y deciden a qué categoría pueden pertenecer. Por ejemplo, para determinar si un correo es Spam o no, se puede usar un clasificador, el cual toma un correo electrónico y predice si es Spam o no. Los algoritmos de clasificación identifican los patrones comunes que hay en los documentos que pertenecen a una clase y usan esta información para clasificar nuevos documentos.

El funcionamiento de los algoritmos de clasificación suelen tener dos etapas según [2] :la etapa de aprendizaje, en la cual se buscan los patrones que hay en común dentro de los documentos de una misma clase; y la etapa de clasificación, en la cual se usan los patrones encontrados para asignar una clase a un nuevo documentos, del cual no se conoce su clase.

Para encontrar los patrones en la etapa de aprendizaje se usa una muestra de los documentos que tienen una clase asignada, esta muestra de documentos es llamado conjunto de entrenamiento. Los elementos de esta muestra pueden ser representados por tuplas con la forma $X = (D,C)$, donde D es un elemento en la fuente de datos (un documento) y C la clase a la que pertenece dicho elemento.

Este tipo de aprendizaje se le conoce como aprendizaje supervisado, dado que cada elemento del conjunto de entrenamiento tiene una clase asignada previamente. Cabe señalar que cuando el conjunto de entrenamiento que no tiene clases asignadas, se denomina como aprendizaje no supervisado [2].

Los patrones encontrados en la etapa de aprendizaje normalmente son representados por reglas de clasificación o modelos matemáticos [2]. En el ejemplo del clasificador de correo Spam, los patrones pueden estar representados por reglas de clasificación, las cuales indicarían cómo identificar a un correo Spam, a partir de los términos que aparecen en el contenido del correo electrónico.

En la etapa de clasificación, primero se evalúa los patrones obtenidos para comprobar su utilidad al análisis. Esta utilidad es evaluada por la precisión que tenga el clasificador al predecir las etiquetas a una muestra de prueba. La muestra de prueba tiene que ser diferente al conjunto de entrenamiento, de lo contrario, podría ocurrir un sobreajuste causando problemas en la clasificación de elementos nuevos. Cabe señalar que la muestra de prueba cuenta con una asignación previa, entonces la precisión se puede obtener comparando estas dos asignaciones, la asignación original y la predicción de hecha por la clasificación.

2.5.1. Árboles de decisión

Los árboles de decisión son una estructura de datos que indica el flujo para llegar a una decisión, sus nodos internos representan una prueba sobre una variable. Cada rama de un nodo interno representa un posible resultado de la prueba. Los nodos que se encuentran al final del árbol tienen una decisión. Estos nodos son conocidos como hojas y el nodo superior del árbol es conocido como raíz o nodo raíz [2].

Supongamos que tenemos a un documento X de la fuente de datos y X no tiene una clase asignada, el árbol de decisión realiza pruebas sobre los términos que tiene X, trazando un camino desde la raíz hasta una hoja del árbol, con este camino se obtiene una decisión que es la predicción de una clase para X.

Los patrones que están representados en el árbol tienen una representación intuitiva y estructurada. El algoritmo que construye al árbol de decisión lo hace de forma recursiva dividiendo el conjunto de entrenamiento, además esta construcción muestra los patrones contenidos en el conjunto de entrenamiento. En [2] se indica que el algoritmo más conocido para crear un árbol de decisión fue desarrollado por J. Ross Quinlan a principios de la década de 1980, y es conocido como ID3 (por sus siglas en inglés "Iterative Dichotomiser") , posteriormente se presentaron mejoras al algoritmo como el C4.5 presentado por Quinlan y el algoritmo CART (por sus siglas en inglés Classification and Regression Trees) publicado por un grupo de estadísticos [6] en 1984.

En el algoritmo ID3, el conjunto de entrenamiento es dividido recursivamente en subconjuntos más pequeños a medida que se construye el árbol.

El algoritmo necesita la muestra de entrenamiento, la lista de las variables que hay en el entrenamiento y una función para la selección de variables, que determina la mejor variable para discriminar las tuplas según su clase.

En el algoritmo 1 que se explicará en breve, se muestra una versión del algoritmo y se describe a continuación. El algoritmo usa un conjunto D que representa al conjunto que está en uso, cuando inicia el algoritmo tiene un solo nodo N y se toma al conjunto de entrenamiento como D (paso 1).

Si en el conjunto D tiene elementos de una sola variable, el nodo N se convierte en una hoja con la etiqueta de esa variable (paso 2 y 3). Cabe señalar que los pasos 4 y 5 son condiciones terminales para la recursión que son explicadas más adelante.

En otro caso, cuando el conjunto D tiene más de una variable, el algoritmo utiliza la función de selección para determinar el criterio de división en D. El criterio de división indica qué variable se usa en el nodo N, para determinar la mejor forma de separar o particionar los elementos en D dentro de clases individuales (paso 6).

El criterio de división determina las ramas que son colocadas en el nodo N tomando en cuenta el resultado obtenido en la prueba. El criterio de división busca que las particiones resultantes para cada rama sean lo más puras como sea posible. Una partición pura tiene elementos con el mismo valor en la variable del criterio de división.

El nodo N es etiquetado con el criterio de división (paso 7) y se coloca una rama desde el nodo N hacia cada uno de los resultados obtenidos por el criterio de división. De esta manera, los elementos en D se dividen con base en dicho criterio (pasos 10 a 11). Sea A la variable de división. A tiene n valores distintos $a_1, a_2, ..., a_n$, de acuerdo con el conjunto de entrenamiento:

1. Si la variable asignada es de valor discreto. Los resultados de la prueba para el nodo N corresponden directamente a los valores que tiene A. Creando una rama por cada uno de los a_j, qué son los valores de la variable A, además se le asigna una etiqueta con ese valor.

   La partición D_j es el subconjunto de elementos etiquetados que pertenecen a D y que tienen el valor a_j de A. Dado que todos los elementos en D_j tiene el mismo valor para A, A no necesita ser considerado en particiones futuras del proceso. Por lo que A es eliminada de la lista de variables (pasos 8 y 9).

2. Si la variable es continua, se crean un punto de división para particionar el conjunto en dos partes $A \leq$ *punto de división* y $A >$ *punto de división*, este punto de división es obtenido por la función de selección como parte del criterio de división. Las dos ramas son colocadas en el nodo N y etiquetas de acuerdo a su respectivo resultado.

A menudo, se toma como punto de división el valor medio de los valores que tiene la variable usada. De esta manera los elementos en D_1 son menores o iguales al punto de división y análogamente los elementos en D_2 tienen valores mayores al punto de división.

3. Si la variable tiene valores binarios se crean dos ramas, la primera tiene asignada la etiqueta "si" o "verdadero" y la otra rama tiene la etiqueta "no" o "falso", la rama "si" representa al subconjunto de los elementos que tienen el valor "si" en a_j de la variable A y análogamente la rama "no" representa a los elementos que tienen el valor "no" en a_j.

El algoritmo continúa con el mismo proceso de forma recursiva para formar el árbol de decisión con los elementos de cada una de las particiones D_j obtenidas de D (paso 14). La partición recursiva se detiene cuando alguna de las siguientes condiciones terminales se cumple:

- Todas las tuplas de la partición D tienen la misma clase (paso 2 y 3).
- No hay variables restantes en los elementos que puedan dividirse más (paso 4), entonces se le asigna al nodo una etiqueta con la variable más común de estos elementos (paso 5), y se convierte en un nodo hoja.
- La partición D_j es vacía (paso 12), en este caso se crea un nodo hoja asignando la clase mayoritaria en D (paso 13).

Y finalmente se obtiene el árbol de decisión generado por el proceso en el paso 15.

```
1  Se crea un nodo N.
2  Si los elementos en el conjunto D tienen el misma variable C, entonces,
3      Regresa a N como un nodo hoja con la variable C.
4  Si la lista de variables está vacía ,
5      Regresa a N como un nodo hoja con la variable mayoritaria del
       conjunto D.
6  Aplica la función de selección sobre el conjunto D y la lista de variables,
     para encontrar el mejor criterio de división.
7  Etiquetar al nodo N usando el criterio de división.
8  Si la variable de división es discreta y la división multiple es posible,
     entonces,
9      Devuelve la lista de variables eliminando la variable de división usada.
10 Para cada subconjunto D_j obtenido de la división, se particiona y crean los
     subárboles de esas divisiones.
11     Sea un D_j en el conjunto de elementos en D que cumplen el criterio de
       separación en j.
12     Si D_j está vacío,
13         Coloca un nodo hoja etiquetado con la variable mayoritaria de D_j.
14     De lo contrario regresa el nodo generado por el algoritmo usando D_j y
       la lista de variables actuales para N.
15 Regresa el nodo N.
```

Algoritmo 1: Muestra el algoritmo básico para la construcción de árboles de decisión usando un conjunto de entrenamiento.

Si usamos un conjunto de entrenamiento D y el algoritmo dado, tenemos que la complejidad computacional está en la fórmula 2.3:

$$O(n * |D| * log(D)) \tag{2.3}$$

Donde n es el número de variables que usan los elementos en D y $|D|$ es el número de elementos en el entrenamiento en D. El costo computacional de construir el árbol es $n * |D| * log|D|$ con $|D|$ elementos.

La función de selección de variables es una heurística que selecciona el criterio de división para hacer la mejor separación para un conjunto de entrenamiento $|D|$, esta separación produce particiones de elementos con el mismo valor de sus variables. Si dividimos el conjunto D en las particiones más pequeñas, siguiendo el criterio de división, idealmente cada partición tendría elementos con las mismas variables. Pero en la práctica es difícil llegar a esa partición, por eso, el mejor criterio de división es el que más se acerque a este escenario.

Las funciones de selección de variables también se conocen como reglas de división porque determinan cómo se dividen los elementos en un nodo.

La función de selección de variables proporciona una puntuación para cada variable que esté presente en el entrenamiento. La variable que tiene la mejor puntuación es seleccionada por la función de selección como la variable de división para los elementos. Las funciones de selección más populares son la ganancia de información, tasa de ganancia e índice de Gini [2].

Tradicionalmente, el algoritmo ID3 usa la ganancia de información como su función de selección de variables. Esta función se basa en el trabajo realizado por Claude Shannon sobre la teoría de la información [2], la cual estudia el valor de la información. La variable con la mayor ganancia de información se elige como la variable de división para el nodo. Esta variable minimiza la información necesaria para clasificar los elementos en las particiones resultantes y refleja la menor aleatoriedad en las particiones.

2.5.2. Clasificadores estadísticos

Los clasificadores bayesianos son clasificadores estadísticos. Estos clasificadores pueden predecir la probabilidades de una tupla que pertenezca a una clase en particular.

Los clasificadores bayesianos ingenuos suponen que el valor de una variable de una tupla es independiente a los otros valores que hay en la tupla. Este supuesto se llama independencia condicional de variables [2] por esta razón se considera como “ingenuo”. Este supuesto permite simplificar los cálculos involucrados. La clasificación bayesiana se basa en el teorema de Bayes. En [2] se menciona que el teorema de Bayes lleva el nombre de Thomas Bayes, que fue un clérigo que trabajó en la teoría de probabilidad y de decisión durante el siglo XVIII.

Sea X una tupla de la fuente de datos, donde X tiene n variables. Sea H una hipótesis que indica que la tupla X pertenece a una clase específica C. En términos de clasificación, se quiere determinar $P(H|X)$, la probabilidad de que la hipótesis H se cumpla dada la “evidencia” o los valores de la tupla X. En otras palabras, se busca la probabilidad de que la tupla X pertenezca a la clase C, dado que se conoce los valores de X.

$P(H|X)$ es la probabilidad posterior o a posteriori, de H condicionada por X. Por ejemplo, supongamos que las tuplas representan a un grupo de clientes de una tienda y los clientes son descritos por las variables edad y los ingresos del cliente. Sea X un

cliente con 35 años y con un ingreso de $40,000. Supongamos que H es la hipótesis de que nuestro cliente comprara una computadora. Luego, $P(H|X)$ refleja la probabilidad de que el cliente X compre una computadora dado que conocemos la edad y los ingresos del cliente.

En contraste, $P(H)$ es la probabilidad posterior o a posteriori, de H. Para nuestro ejemplo, esta es la probabilidad de que un cliente en específico compre una computadora, independientemente de su edad, ingresos o cualquier otra información. La probabilidad posterior $P(H|X)$ se basa en más información que en la probabilidad a priori $P(H)$ que es independiente de X.

De manera similar, $P(X|H)$ es la probabilidad posterior de X condicionada por H. Es decir, es la probabilidad de que un cliente X, tenga 35 años y gane $40,000, dado que sabemos que el cliente compró una computadora. $P(X)$ es la probabilidad previa de X. Usando nuestro ejemplo, es probable que una persona de nuestro grupo de clientes tenga 35 años y gane $40,000. La estimación de las probabilidades $P(H)$, $P(H|X)$ y $P(X)$ se obtiene a partir de las tuplas que hay en la fuente de datos. El teorema de Bayes es útil porque proporciona una forma de calcular la probabilidad posterior $P(H|X)$,a partir de $P(H)$,$P(H|X)$ y $P(X)$, el teorema de Bayes se muestra en la ecuación 2.4.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \tag{2.4}$$

A continuación, se muestra el funcionamiento del clasificador Bayesiano ingenuo:

1. Sea D un conjunto de tuplas de un conjunto de entrenamiento de una fuente de datos, cada tupla con su clase asociada. Las tuplas están representadas por un vector de variables $X = x_1, x_2, ..., x_n$, donde cada variable puede tomar un valor A_i para la variable x_i.

2. Supongamos que hay m clases $C_1, C_2, ..., C_m$ en el conjunto de entrenamiento, Sea X una tupla, el clasificador predecirá que X pertenece a la clase que tiene la probabilidad posterior condicionada más alta para X. Es decir, el clasificador bayesiano predice que la tupla X pertenece a la clase C_i si y sólo si:

$$P(C_j|X) \quad con \quad 1 \leq j \leq m, \quad j \neq i \tag{2.5}$$

3. Como $P(X)$ es constante para todas las clases, sólo se necesita maximizar $P(X|C_i) * P(C_i)$. Si no se conocen las probabilidades anteriores de la clase, se supone comúnmente que las clases son igual de probables, es decir, $P(C_1) = P(C_2) = ... = P(C_m)$ y así se maximiza solo $P(X|C_i)$. De lo contrario se maximiza $P(X|C_i)P(C_i)$. Cabe señalar que las probabilidades previas de la clase pueden estimarse con $P(C_i) = \frac{|C_{i,D}|}{|D|}$, donde $|C_i, D|$ es el número de tuplas en el entrenamiento de la clase C_i en D.

4. Si el conjunto de entrenamiento tiene muchas variables, sería computacionalmente costoso calcular $P(X|C_i)$. Para reducir el cálculo en la evaluación de $P(X|C_i)$, se hace la suposición ingenua de la independencia condicional. Esto supone que los valores de las variables son condicionalmente independientes entre sí, sin importar que clase tiene asignada la tupla , es decir, no hay relaciones de dependencia entre las variables, en consecuencia se tiene:

$$P(X|C_i) = \prod_{k_i=1}^{n} P(x_k|C_i) \tag{2.6}$$

$$= P(x_1|C_i) \quad X \quad P(x_2|C_i) \quad X...X \quad P(x_n|C_i) \tag{2.7}$$

Podemos estimar fácilmente las probabilidades de $P(x_1|C_i), P(x_2|C_i), ..., P(x_n|C_i)$ a partir de las tuplas de entretenimiento. Recordando que x_k se refiere al valor de la variable A_k para la tupla X. Cada variable puede ser categórica o de valor continuo. Para calcular $P(X|C_i)$, se considera lo siguiente:

a) Si A_k es categórica, entonces $P(x_k|C_i)$ es el número de elementos de la clase C_i en D que tiene el valor x_k para A_k, dividido por $|C_i, D|$, que es el número de elementos de la clase C_i en D.

b) Si A_k es de valor continuo, se suele suponer que una variable de valor continuo tiene una distribución gaussiana con una media μ y una desviación estándar σ.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \tag{2.8}$$

de modo que,

$$P(x_k|C_i) = g(x_k, \mu_{Ci}, \sigma_{Ci}) \tag{2.9}$$

Para calcular C_i y C_i, se utiliza el promedio y la desviación estándar respectivamente de los valores de la variable A_k para entrenar las tuplas de la clase C_i.

Luego usamos estas dos cantidades, junto con x_k, para estimar $P(x_k|C_i)$.

5. Para predecir la clase de X, se evalúa $P(X|C_i)P(C_i)$ para cada clase C_i. El clasificador predice la clase de X, la cuál es la clase C_i si y sólo si:

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad para \quad 1 \leq j \leq m, j \neq i. \tag{2.10}$$

En otras palabras, la clase predicha para X es C_i, donde se cumple que $P(X|C_i)P(C_i)$ es la máxima probabilidad.

En estudios empíricos para medir la eficiencia de este clasificador, se obtuvieron resultados comparables con los clasificadores que usan en un árbol de decisión y redes neuronales. En teoría, los clasificadores bayesianos tienen la tasa de error mínima en comparación con todos los demás clasificadores [2]. Sin embargo, en la práctica no es siempre el caso, debido a las suposiciones hechas en su uso, como la independencia condicional de las variables y la falta de datos disponibles sobre la probabilidad.

Los clasificadores bayesianos son útiles porque proporcionan una justificación teórica, ya que hay otros clasificadores que no utilizan explícitamente el teorema de Bayes, además también han demostrado alta precisión y velocidad cuando se aplican a grandes cantidades de datos. Por ejemplo, bajo ciertos supuestos, se puede demostrar que muchos algoritmos de ajuste de curvas y redes neuronales, generan la hipótesis de máxima probabilidad posterior, al igual que el clasificador bayesiano ingenuo [2].

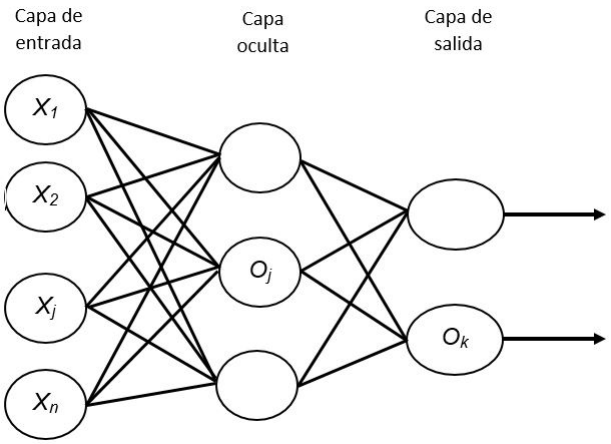


Figura 2.3: Muestra una red neuronal de alimentación multicapa.

2.5.3. Redes Neuronales

En el campo de las redes neuronales fue originado por psicólogos y neurobiólogos que buscaban desarrollar y probar modelos computacionales análogos a las neuronas. En términos generales, una red neuronal es un conjunto de unidades de entrada y salida conectadas entre sí, en la que cada conexión tiene un peso asociado.

El algoritmo de retropropagación ajusta los pesos de la red procesando iterativamente un conjunto de tuplas de entrenamiento, comparando la predicción de la red con el valor real conocido de cada una de las tuplas.

La red neuronal de alimentación de múltiples capas consta de una capa de entrada, una o más capas ocultas y una capa de salida, en la figura 2.3 se muestra la red de alimentación de múltiples capas.

Cada capa está compuesta de varias unidades. Las entradas a la red corresponden a las variables de cada tupla de entrenamiento. Estas entradas alimentan simultáneamente a las unidades de la capa de entrada. Las entradas pasan a través de la capa de entrada y luego son ponderadas y alimentan simultáneamente a una segunda capa de unidades, conocida como capa oculta. La salida de las unidades que hay en la capa oculta se pueden ingresar a otra capa oculta, y así sucesivamente. Las salidas ponderadas de la última capa oculta se ingresan a las unidades que forman la capa de salida, las cuales emiten la predicción de la red para un elemento dado.

Las unidades en las capas se denominan neuronas, debido a su base biológica simbólica. La red neuronal multicapa que se muestra en la figura 2.3 tiene dos capas, o como es conocida, una red neuronal de dos capas [2]. La capa de entrada no se cuenta porque solo sirve para pasar los valores de entrada a la siguiente capa. De manera similar, una red que contiene dos capas ocultas sería una red neuronal de tres capas, y así sucesivamente. La red está fuertemente conectada porque cada capa proporciona una entrada a cada unidad de la siguiente capa “hacia adelante”. Es una red de propagación hacia adelante, ya que ninguna de las conexiones regresa a una unidad de una capa anterior.

Cada unidad de salida toma como entrada una suma ponderada de las salidas de las unidades en la capa anterior. Aplicando una función de activación ponderada. Las redes

neuronales de alimentación multicapa pueden modelar la predicción de una clase como una combinación no lineal de las entradas. Desde un punto de vista estadístico, realizan una regresión no lineal. Las redes de alimentación de múltiples capas con varias unidades ocultas y varias muestras de entrenamiento, pueden aproximarse a cualquier función [2].

Antes de iniciar con el entrenamiento de la red, es necesario decidir la topología de la red especificando la cantidad de unidades en la capa de entrada, la cantidad de capas ocultas con sus respectivas unidades y el número de unidades en la capa de salida.

La normalización de los valores de entrada para cada variable en las tuplas de entrenamiento ayudará a acelerar la fase de aprendizaje. Por lo general, los valores de entrada se normalizan entre 0.0 y 1.0 . Las variables con valores discretos pueden codificarse de modo que haya una unidad de entrada por cada valor de esa variable. Por ejemplo, si una variable A tiene tres valores posibles (a_0, a_1, a_2), entonces podemos asignar tres unidades de entrada para representar a A. Es decir, podemos tener I_0,I_1,I_2, como unidades de entrada. Cada unidad se inicializa en 0, Si A=a_0 entonces I_0 se establece con 1 y el resto son 0. Si A=a_1 entonces I_1 se establece en 1 y el resto son 0, y así sucesivamente.

Las redes neuronales se pueden usar para la clasificación, es decir, predecir la clase de un elemento o un documento; así mismo se pueden usar para una predicción numérica, o para predecir una salida de valor continuo. Para la clasificación, se puede usar una unidad de salida para representar dos clases. Si hay más de dos clases, se usa una unidad de salida por clase.

Durante la fase de aprendizaje, la red ajusta los pesos para poder predecir la clase correcta para cada uno de los elementos de entrada. El aprendizaje de redes neuronales también se conoce como aprendizaje conexionista debido a las conexiones que hay entre unidades.

Hay muchos tipos diferentes de redes neuronales y algoritmos de aprendizaje de redes neuronales. El algoritmo de aprendizaje más popular es la retropropagación, que ganó reputación en la década de 1980 [2]. El algoritmo de propagación hacia adelante realiza el aprendizaje en una red neuronal de alimentación multicapa. Aprende iterativamente de un conjunto de entrenamiento para ajustar los pesos, que servirán para predecir la clase de las elementos.

El resultado esperado es la predicción de una clase o una etiqueta del conjunto de entrenamiento para problemas de clasificación, o un valor continuo para predicción numérica. Para cada tupla de entrenamiento, los pesos se modifican para minimizar el error cuadrático medio entre la predicción de la red y el valor real esperado. Estas modificaciones se realizan en la dirección “hacia atrás”, desde la capa de salida a través de cada oculta hasta la primera capa oculta, de ahí el nombre de retropropagación o propagación hacia atrás. Aunque no está garantizado, en general, los pesos eventualmente convergen y el proceso de aprendizaje se detendrá. Esto se resume en el algoritmo 2. Los pasos involucrados se expresan en términos de entradas, salidas y errores. Los pasos se describen a continuación.

Entradas: Un conjunto D de datos que consiste en los elementos de entrenamiento y sus valores objetivo asociados, l la tasa de aprendizaje. Una red de alimentación multicapa.
Inicializar todos los pesos y sesgos en la red;
Mientras la condiciones de término no sean cumplidas {
Para cada elemento X en D {
// Propaga las entradas hacia adelante:
Para cada unidad de capa de entrada j {
O_j=I_j // la salida de una unidad de entrada es el valor real que tiene la entrada
Para cada unidad en la capa oculta o de salida j {
$I_j = \sum_{i=1} w_{ij} O_i + \theta_j$ //calcula la entrada neta de la unidad j con respecto a la capa anterior i
$O_j = \frac{1}{1+e^{-Ij}}$, // calcula la salida de cada unidad j
// Volver a propagar los errores:
Para cada unidad j en la capa de salida
$Err_j = O_j(1-O_j)(T_j-O_j)$// calcula el error
Para cada unidad j en las capas ocultas, desde la última hasta la primera capa oculta.
$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$, // calcula el error con respecto a la siguiente capa superior k
Por cada peso w_{ij} en red {
$\triangle w_{ij} = (l) Err_j O_i$, // incremento de peso
w_{ij}=$w_{ij} + w_{ij}$, // actualiza los pesos
Para cada sesgo j en la red {
$\triangle \theta_j = (l) Err_j$, // incrementa el sesgo
$\triangle w_{ij} = w_{ij} + \triangle w_{ij}$, } // actualiza el sesgo
}}

Algoritmo 2: Muestra los pasos para el algoritmo de retropropagación.

Inicialización de los pesos: los pesos en la red se inician en números aleatorios. Cada unidad tiene un sesgo asociado, como se explica más adelante. Los sesgos se inician de manera similar con números aleatorios, que se ve reflejado en el paso 2. Cada elemento X del conjunto de entrenamiento, se procesa mediante los siguientes pasos.

Propagación de las entradas hacia adelante: primero, un elemento del conjunto entrenamiento alimenta a la capa de entrada de la red. Las entradas pasan a través de las unidades de entrada, sin cambios, es decir, para una unidad de entrada j, su salida O_j es igual a su valor de entrada I_j. A continuación, se calculan las salidas de cada capa oculta como una combinación lineal de sus entradas, con cada uno de sus pesos formando una suma ponderada, agregando un sesgo asociado a la unidad j y aplicando una función de activación a la entrada. Las entradas de la unidad j son de la forma y_1 ,y_2, ..., y_n. Si la unidad j estuviera en la primera capa oculta, estas entradas corresponden al elemento de entrada (x_1, x_2, ... , x_n), se muestra una capa oculta en la figura 2.4.

Cada unidad tiene una cantidad de entradas que son las salidas de las unidades conectadas a ella en la capa anterior. Cada conexión tiene un peso, para calcular la entrada neta de la unidad, se multiplica cada entrada conectada a la unidad por su peso correspondiente, y esto se suma. Dada una unidad j de una capa oculta o de salida, la

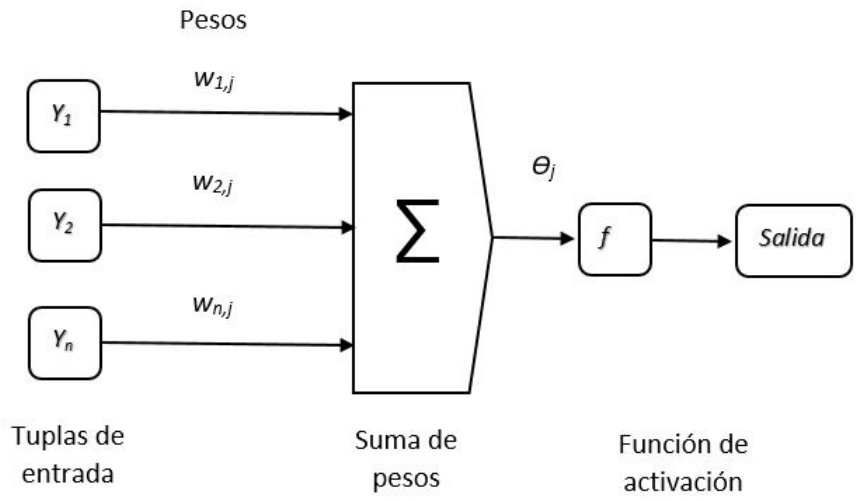


Figura 2.4: La unidad de capa oculta o de salida j. Las entrada de la unidad j son salidas de la capa anterior.

entrada neta es I_j, entonces para la unidad j se tiene:

$$I_j = \sum_{i=1} w_{ij} O_i + \theta_j \tag{2.11}$$

Donde w_{ij} es el peso de la conexión desde la unidad i de la capa anterior en la unidad j, O_i es la salida de la unidad i de la capa anterior y j es el sesgo de la unidad. El sesgo actúa como un umbral que sirve para variar la actividad de la unidad.

Cada unidad en las capas ocultas y de salida toma su entrada neta y luego le aplica una función de activación, como se ilustra en el algoritmo 2. La función simboliza la activación de la neurona representada por la unidad. Se utiliza la función logística o sigmoidea. Dada la entrada I_j a la unidad j, entonces O_j es la salida de la unidad j, y se calcula como:

$$O_j = \frac{1}{1 + e^{-I_j}} \tag{2.12}$$

Esta función también se conoce como una función de aplastamiento, ya que asigna un gran dominio de entrada en el rango más pequeño de 0 a 1. La función logística es no lineal y diferenciable, lo que permite que el algoritmo de retropropagación modele problemas de clasificación que son no lineales. Se calculan los valores de salida O_j, para cada capa oculta hasta la capa de salida, la cual es la capa que proporciona la predicción de la red. En la práctica, es una buena idea almacenar los valores de salidas intermedios de cada unidad, ya que se requieren nuevamente más tarde cuando se propaga el error. Este truco puede reducir sustancialmente la cantidad de cálculo requerido, donde este proceso se puede ver reflejado en los pasos 4 al 10.

Propagar el error hacia atrás: el error se propaga hacia atrás actualizando los pesos y sesgos para reflejar el error de la predicción de la red. Para una unidad j en la

capa de salida, el error Err_j es calculado por:

$$Err_j = O_j(1-O_j)(T_j-O_j) \tag{2.13}$$

Donde w_{jk} es el peso de la conexión de la unidad j a una unidad k en la siguiente capa superior y Err_k es el error de la unidad k.

Los pesos y sesgos se actualizan para reflejar los errores propagados. Los pesos se actualizan mediante las siguientes ecuaciones, donde $\triangle w_{ij}$ es el cambio en el peso w_{ij}.

$$\triangle w_{ij} = (l)Err_j O_i \tag{2.14}$$

$$\triangle w_{ij} = w_{ij} + \triangle w_{ij} \tag{2.15}$$

La variable l es la tasa de aprendizaje, una constante que típicamente tiene un valor entre 0.0 y 1.0. La retropropagación aprende utilizando un método de descenso de gradiente para buscar un conjunto de pesos que se ajuste a los datos de entrenamiento para minimizar la distancia media entre la predicción de clase de la red y el valor conocido del conjunto de entrenamiento. La tasa de aprendizaje ayuda a evitar quedarse atascado en un mínimo local en el espacio de decisión, donde los pesos parecen que convergen, pero no son la solución óptima y esto alienta a encontrar el mínimo global. Si la tasa de aprendizaje es demasiado pequeña, entonces el aprendizaje ocurrirá a un ritmo muy lento. Si la tasa de aprendizaje es demasiado grande entonces los pesos pueden tardar en converger en una solución ideal . Una regla general es establecer la tasa de aprendizaje en $1/t$, donde t es el número de iteraciones a través del conjunto de entrenamiento hasta el momento.

Los sesgos se actualizan mediante las siguiente ecuaciones, donde $\triangle \theta_j$ es el cambio de sesgo θ_j:

$$\triangle \theta_j = (l)Err_j \tag{2.16}$$

$$\theta_j = \theta_j + \triangle \theta_j \tag{2.17}$$

Aquí se están actualizando los pesos y sesgos después de la presentación de cada elemento. Esto se conoce como actualización de los casos. Alternativamente, los incrementos de peso y sesgo podrían acumularse en variables, de modo que los pesos y sesgos se actualicen después de que se hayan presentado todo el conjunto de entrenamiento. Esta última estrategia se llama época, la cual utilizó todos los elementos de un conjunto del entrenamiento. En teoría, la derivación matemática de la retropropagación emplea las épocas, pero en la práctica, la actualización de casos es más común porque tiende a producir resultados más precisos, está parte del algoritmo está reflejado en los pasos 11 al 22.

Condiciones de termino; el entrenamiento se detiene cuando:

- Todos los $\triangle w_{ij}$ en la época anterior son tan pequeños como para estar por debajo de un umbral específico, o
- El porcentaje de los elementos mal clasificados en la época anterior está por debajo de algún umbral, o
- Ha alcanzado el número predeterminado de épocas.

En la práctica, se pueden requerir varios cientos de miles de épocas antes de que los pesos converjan. La eficiencia computacional de la retropropagación depende del tiempo dedicado a entrenar la red. Dadas las tuplas $|D|$ y los pesos w, cada época requiere $O(|D| * w)$ de unidades de tiempo. Sin embargo, en el peor de los casos, el número de épocas puede ser exponencial al número de entradas n. Existen varias técnicas que ayudan a acelerar el tiempo de entrenamiento, como recocido simulado, que también asegura la convergencia a un óptimo global.

En [2] menciona que es difícil para los humanos interpretar el significado simbólico detrás de los pesos aprendidos y de las "unidades ocultas" en la red. Porque las redes tienen un funcionamiento de "caja negra", lo cual resulta complicado para el análisis, ya que no da información explícita de las relaciones creadas. Estas características inicialmente hicieron que las redes neuronales fueran menos deseables para la minería de datos. Sin embargo, las ventajas de las redes neuronales incluyen su alta tolerancia a los datos ruidosos, así como su capacidad para clasificar patrones, los cuales no aparecen en el entrenamiento. Se pueden usar cuando se tenga poco conocimiento de las relaciones entre variables y clases. Son muy adecuadas para entradas y salidas de valor continuo, a diferencia de la mayoría de los algoritmos de árboles de decisión.

2.5.4. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM por sus siglas en inglés *Support Vector Machine*) son un método para la clasificación de datos lineales y no lineales. Una SVM es un algoritmo que funciona de la siguiente manera. Utiliza un mapeo no lineal para transformar los datos de entrenamiento originales a una dimensión superior. Dentro de esta nueva dimensión, busca el hiperplano de separación lineal óptimo, el cual separa las tuplas de una clase de otra. Con un mapeo no lineal apropiado en una dimensión suficientemente alta, los datos de dos clases siempre pueden ser separados por un hiperplano. Las SVM encuentran este hiperplano utilizando vectores de soporte, estos vectores de soporte son elementos del conjunto de entrenamiento "esenciales" para realizar la clasificació.

El primer artículo sobre máquinas de soporte vectorial fue presentado en 1992 por Vladimir Vapnik y sus colegas Bernhard Boser e Isabelle Guyon, aunque las ideas preliminares vienen desde la década de 1960 [2]. Las SVM se pueden usar para la predicción numérica y la clasificación. Se han aplicado a una serie de áreas, incluido el reconocimiento de dígitos a mano, el reconocimiento de objetos y el reconocimiento de voz. De acuerdo con [2], son mucho menos propensos al sobreajuste en comparación a otros métodos.

Clasificación de datos linealmente separables.

Considerando un caso simple de clasificación usando las SVM, con un problema de dos clases, donde las clases son linealmente separables. Dado un conjunto de datos D, como (X_1^j, Y_1), (X_2^j, Y_2), ..., $(X_{|D|}^j, Y_{|D|})$, donde (X_i^j) es el conjunto de elementos de entrenamiento con su clase asociada Y_i. Cada Y_i puede tomar uno de los dos valores $+1$ o -1, es decir, $Y_i \in \{+1, -1\}$.

Los datos de entrenamiento en 2D mostrados en la figura 2.5, son linealmente separables. Y hay un número infinito de posibles hiperplanos de separación o "límites de decisión", algunos de los cuales se muestran como líneas punteadas.

Consideremos un ejemplo basado en dos variables de entrada, $A1$ y $A2$, como se muestra en la figura 2.5, vemos que los datos en 2D son linealmente separables, porque

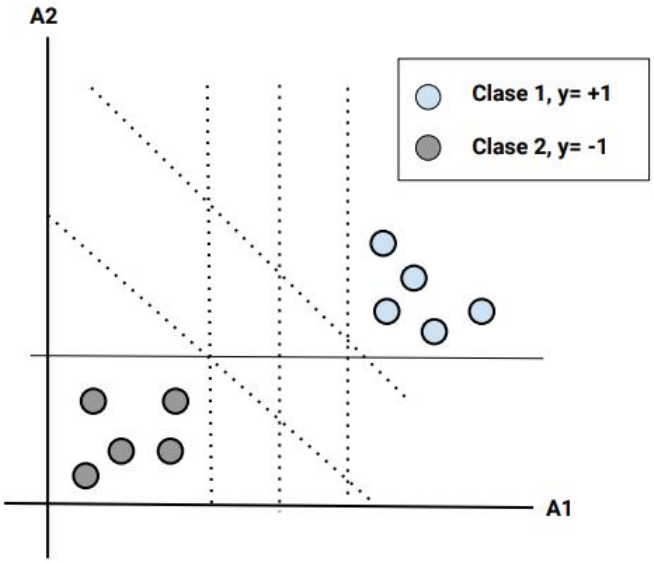


Figura 2.5: Muestra los posibles hiperplanos de separación o ”límites de decisión”, que pueden ser encontrados.

se puede dibujar una línea recta para separar todas las tuplas de la clase $+1$ de todas las tuplas de la clase -1.

Hay un número infinito de líneas de separación que podrían dibujarse. Queremos encontrar la ”mejor”, es decir, una que tenga el mínimo error de clasificación en las tuplas. Para encontrar la mejor línea hay que tener en cuenta nuestros datos, ya que si fueran tridimensionales, es decir, con tres variables, el problema se convierte en encontrar el mejor plano de separación. Generalizando esta idea a n dimensiones, debemos encontrar el mejor hiperplano independientemente del número de variables de entrada.

En la figura 2.6 se muestra dos posibles hiperplanos de separación y sus márgenes asociados. Ambos hiperplanos pueden clasificar correctamente todas las tuplas de datos. Sin embargo, intuitivamente, esperamos que el hiperplano con el margen más grande sea más preciso para clasificar futuras tuplas de datos en comparación al margen más pequeño. Es por eso que durante la fase de aprendizaje o entrenamiento, la SVM busca el hiperplano con el margen más grande, es decir, el hiperplano de margen máximo (MMH por sus siglas en inglés **Maximum Marginal Hyperplane**). El margen asociado proporciona la mayor separación entre clases.

Una definición informal de margen, puede ser que la distancia más corta desde un hiperplano a un lado de su margen es igual a la distancia más corta desde el hiperplano al otro lado de su margen, donde estos ”lados”del margen son paralelo al hiperplano. Cuando se trata con el MMH, esta distancia es de hecho la distancia más corta desde el MMH hasta el elemento del conjunto de entretenimiento más cercano de cualquier clase. Un hiperplano de separación se puede escribir como:

$$WX + b = 0 \tag{2.18}$$

Donde W es un vector de peso, $W = (w_1, w_2, ..., w_n)$, n es el número de variables y con b un escalar, a menudo denominado sesgo. Para ayudar en la visualización,

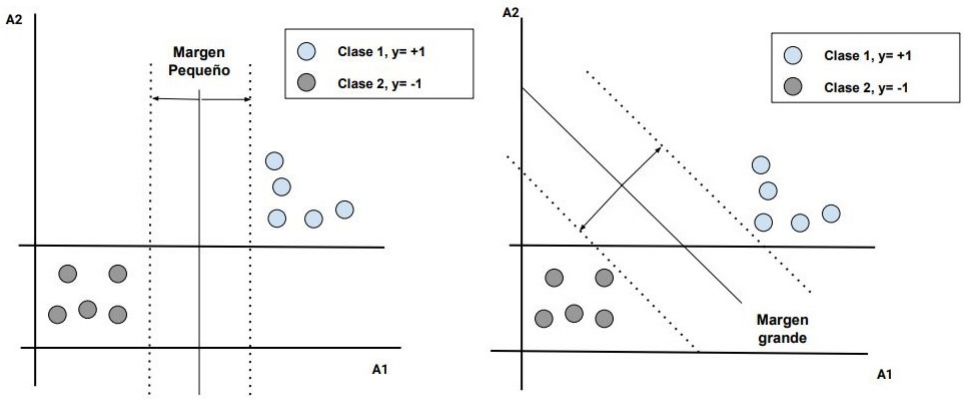


Figura 2.6: Se muestran dos posibles hiperplanos de separación y sus márgenes asociados. El margen grande tiene una mayor precisión.

consideremos dos variables de entrada A_1 y A_2, como en la figura 2.5. Las tuplas de entrenamiento son 2D, por ejemplo $X = (x_1, x_2)$, donde x_1 y x_2 son los valores de las variables A_1 y A_2, respectivamente, para X. Si pensamos en b como un peso adicional w_0, podemos reescribir la ecuación como:

$$w_0 + w_1 x_1 + w_1 w_2 = 0 \tag{2.19}$$

Por lo tanto, cualquier punto que se encuentre por encima del hiperplano de separación satisface.

$$w_0 + w_1 x_1 + w_2 x_2 > 0 \tag{2.20}$$

Los pesos se pueden ajustar para que los hiperplanos que definen los lados del margen se puedan escribir como:

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1 \qquad para \qquad y_i = 1 \tag{2.21}$$

$$H_2 : w_0 + w_1 x - 1 + w_2 x_2 \leq 1 \qquad para \qquad y_i = -1 \tag{2.22}$$

Cualquier tupla que cae sobre H_1 o por encima pertenece a la clase $+1$, y las tuplas que caen sobre o por debajo de H_2 pertenecen a la clase -1. Combinando las dos desigualdades H_1 y H_2, obtenemos:

$$y_i(w_0 + w_1 x_1 + w_2 x_2) \geq 1 \qquad para\ toda \qquad i. \tag{2.23}$$

Cualquier elemento del conjunto de entrenamiento que caiga en los hiperplanos H_1 o H_2, es decir, los lados que definen el margen y que satisfacen la ecuación 2.23 se denominan vectores de soporte. Es decir, están igualmente cerca del MMH que los separa. En la figura 2.7 los vectores de soporte se muestran rodeados con un borde más grueso. Esencialmente, los vectores de soporte son los elementos más difíciles de clasificar y brindan la mayor información con respecto a la clasificación.

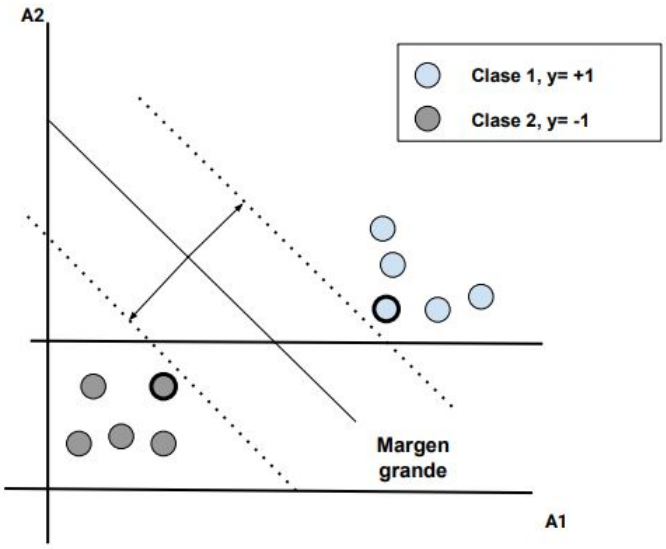


Figura 2.7: Se muestra el hiperplano de separación máximo.

A partir de esto, podemos obtener una fórmula para el tamaño del margen máximo. La distancia desde el hiperplano para separar a cualquier punto en H_1 es $\frac{2}{||W||}$, donde $||W||$ es la norma euclidiana de W, es decir, $\sqrt{(W \cdot W)^2}$. Por definición, esto es igual a la distancia desde cualquier punto a H_2 el hiperplano de separación. Por lo tanto, el margen máximo es $\frac{2}{||W||}$.

La SVM encuentra el hiperplano de separación máximo, es decir, el que tiene la distancia máxima entre las tuplas de entrenamiento más cercanas.

En [2] plantea que los vectores de soporte se encuentran reescribiendo la ecuación 2.23, para que se convierta en un problema conocido como problema de optimización cuadrática restringido (convexo). Usando una función lagrangiana y luego resolviendo la solución usando condiciones de Karush-Kuhn-Tucker (KKT).

En [2] menciona que si la fuente de datos tiene menos de 2000 tuplas de entrenamiento, cualquier paquete de software de optimización para resolver problemas cuadráticos convexos restringidos puede usarse para encontrar los vectores de soporte y MMH. Para datos más grandes, se puede utilizar algoritmos especiales y más eficientes para entrenar a una SVM.

Cuando se obtienen los vectores de soporte y por consiguiente el MMH, se puede considerar que la SVM ya está entrenada, y la SVM es capaz de clasificar conjuntos de datos linealmente separables. Para clasificar nuevos elementos con la SVM se usa la formulación lagrangiana mencionada anteriormente, el MMH puede reescribirse como:

$$d(X^T) = \sum_{i=1}^{l} y_i \alpha_i X_i X^T + b_0 \tag{2.24}$$

Donde y_i es la clase del vector de soporte X_i, X^T es un elemento de prueba, α_i y b_0 son parámetros numéricos que fueron determinados automáticamente por el algoritmo de optimización o la SVM mencionado anteriormente y l es el número de vectores de

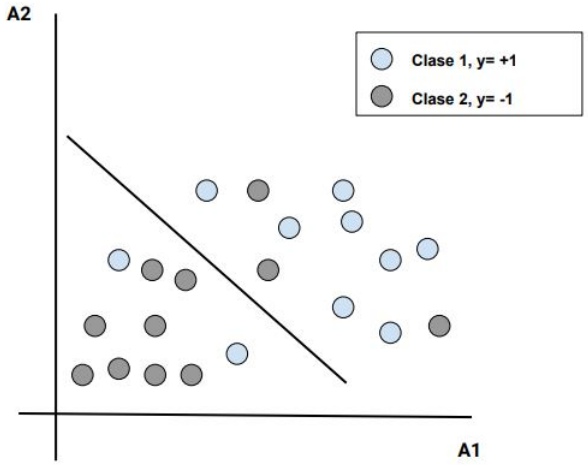


Figura 2.8: Muestra un ejemplo 2-D no separable linealmente.

soporte. Cabe señalar que α_i es un multiplicador lagrangiano.

Clasificación de datos linealmente no separables

En la figura 2.8 se muestra un ejemplo en 2D con datos linealmente no separables. A diferencia de los datos separables lineales de la figura 2.5 aqui no es posible dibujar una línea recta para separar las clases. En este ejemplo el límite de decisión es no lineal.

Cuando se tienen conjuntos linealmente no separables, estos no pueden ser clasificados por las SVM lineales que anteriormente se describieron. Para estos conjuntos se extienden las SVM lineales para crear SVM no lineales, las cuales, son capaces de clasificar datos linealmente no separables, también llamados datos linealmente no separables o datos no lineales [2]. Tales SVM no lineales son capaces de encontrar límites de decisión no lineales.

Para extender una SVM lineal a una SVM no lineal se procede de la siguiente manera. Hay dos pasos principales: el primer paso es transformar los datos de entrada originales en un espacio dimensional superior utilizando una transformación no lineal; después en el segundo paso, se busca el hiperplano de separación lineal en el nuevo espacio. Esto nos lleva a un problema de optimización cuadrática que se puede resolver utilizando una SVM lineal . El hiperplano marginal máximo encontrado en el nuevo espacio corresponde a una hipersuperficie de separación no lineal en el espacio original.

Para obtener la transformación no lineal de los datos de entrada originales en un espacio dimensionalmente superior, se sigue el siguiente ejemplo.

Sea un vector de tres dimensiones $X = (x_1, x_2, x_3)$, lo llevaremos a un espacio de seis dimensiones, llamado (Z), usando la siguiente transformación, $\varphi_1(X) = x_1$, $\varphi_2(X) = x_2$, $\varphi_3(X) = x_3$, $\varphi_4(X) = x_1^2$, $\varphi_5(X) = x_1x_2$ y $\varphi_6(X) = x_1x_3$. Obteniendo un hiperplano de decisión en el nuevo espacio definido por $d(Z) = WZ + b$, donde W y Z son vectores.

Resolvemos W, b y luego sustituimos de nuevo para que el hiperplano de decisión lineal en el nuevo espacio (Z) corresponda a un polinomio no lineal de segundo orden

en el espacio de entrada 3D original.

$$d(Z) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1^2 + w_5x_1x_2 + w_6x_1x_3 + b. \tag{2.25}$$

$$= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6x_6 + b. \tag{2.26}$$

Pero con esta solución se pueden originar algunos problemas relacionados con el tiempo de ejecución. Como el tiempo que toma en calcular varias veces el producto punto de los elementos de entrenamiento para encontrar MMH y el tiempo que toma calcular los elementos de prueba con los vectores de soporte.

Este problema se puede resolver usando la optimización cuadrática de la SVM lineal, es decir, buscando una SVM lineal para el nuevo espacio dimensionalmente superior, donde las tuplas de entrenamiento aparecen en la forma de productos punto como $\varphi(X_i) \cdot \varphi(X_j)$, recordando que $\varphi(X)$ es la función de transformación no lineal aplicada a las tuplas de entrenamiento. Ahora, en lugar de calcular el producto punto de las tuplas de los datos transformados, resulta que es matemáticamente equivalente aplicar una función $K(X_i, X_j)$, a los datos de entrada originales, es decir:

$$K(X_i, X_j) = \varphi(X_i) \cdot \varphi(X_j) \tag{2.27}$$

En otras palabras, donde aparece $\varphi(X_i) \cdot \varphi(X_j)$ dentro del algoritmo de entrenamiento, podemos reemplazarlo por $K(X_i, X_j)$, que es conocida como función núcleo. De esta forma, todos los cálculos se realizan en el espacio original, el cual tiene una dimensión mucho menor. Con esto podemos encontrar un hiperplano de separación máxima. El procedimiento es similar al anterior, aunque implica colocar un límite superior especificado, en los multiplicadores de Lagrange, *i*. El límite superior se determina mejor experimentalmente según [2].

En [2] se proponen tres funciones núcleo, las cuales son:

- **Núcleo polinomial de grado h**: $K(X_i, X_j) = (X_i \cdot X_j + 1)^h$.
- **Núcleo de base radial gaussiana**: $K(X_i, X_j) = e^{-||X_i - X_j||^{\frac{2}{2\sigma^2}}}$.
- **Núcleo sigmoide**: $K(X_i, X_j) = tanh(kX_i \cdot X_j - \delta)$.

Una SVM con un núcleo sigmoide es equivalente a una red neuronal de dos capas conocida como perceptrón multicapa. No hay reglas para determinar qué núcleo dará la SVM con mayor precisión. En la práctica, el núcleo elegido generalmente no hace una gran diferencia en la precisión resultante. Una SVM entrenada siempre encuentra una solución global en problemas donde existen muchos mínimos locales, a diferencia de las redes neuronales de retropropagación [2].

Los vectores de soporte son los elementos de entrenamiento esenciales o críticas, recordando que estas tuplas son las que se encuentran más cerca al límite de decisión o MMH. Si se eliminaran todas las tuplas que no son parte de los vectores de soporte y se repite el entrenamiento, se encontrara el mismo hiperplano de separación. Además, el número de vectores de soporte encontrados se pueden usar para calcular un límite superior en la tasa de error esperada del clasificador de SVM, y este límite es independiente de la dimensionalidad de los datos [2].

Una SVM es independiente de la dimensionalidad de los datos y una SVM con un pequeño número de vectores de soporte puede tener una buena generalización, incluso cuando la dimensionalidad de los datos es alta. Un objetivo de investigación importante

con respecto a las SVM es mejorar la velocidad en el entrenamiento y las pruebas para que las SVM se conviertan en una opción más factible para conjuntos de datos muy grandes .Y otros problemas incluyen determinar el mejor núcleo para un conjunto de datos.

Resumen

En este capítulo se menciona que es la minería de datos y su analogía con la minería de textos, además que ese explican el funcionamiento de cada una de las etapas que son: tareas de preprocesamiento que buscan eliminar caracteres y las palabras irrelevantes para el análisis, los algoritmos de minería de texto, los cuales buscan los patrones para clasificar a nuevos documentos, otra tarea es la visualización de la información que muestra la información obtenida de una manera sencilla y rápida, por último es el refinamiento, este consiste en aplicar de nuevo alguna tarea previa pero con un cambio en la configuración. Esta última tarea se aplica para obtener el mejor resultado posible.

Además se explican el funcionamiento de los principales algoritmos de minería de textos, los cuales son: árboles de decisión, clasificadores probabilísticos, redes neuronales y máquinas de soporte vectorial.

Capítulo 3

Análisis de cuestionarios

La investigación del aprendizaje de las ciencias ha sufrido varios cambios a través del tiempo, esto ha llevado a desarrollar nuevos enfoques y nuevas formas de enseñanza de las ciencias, las cuales han implicado varias transformaciones en los procesos e instrumentos de investigación y de evaluación. Una de estas transformaciones en los instrumentos de investigación, plantea centrarse en el estudio e identificación de los conceptos alternativos o ideas previas que tienen los estudiantes, para generar nuevas metodologías e instrumentos que nos permitan evaluar el aprendizaje y comprensión de la ciencia que tienen los estudiantes.

Por ejemplo, en [1] se indica que el aprendizaje de la genética es poco significativo y escasamente comprendido por los estudiantes, ya que hay estudiantes que muestran dificultades para comprender y representar los conceptos relacionados, adicionalmente, se indica que esto es un reflejo de una problemática relacionada con la forma actual de enseñanza de las ciencias. También se indica que, en el tema de genética es muy común el uso de representaciones externas, ya que esto permite a los profesores puedan presentar los conceptos y sus relaciones de una manera más clara. Por ejemplo, empleando imágenes para la representación de un cromosoma o ADN como apoyo a su explicación verbal. En contraste, la mayoría de los instrumentos diseñados para evaluar el conocimiento de los alumnos, no permiten que el estudiante haga uso de diferentes representaciones externas para elaborar su respuesta; además que con los instrumentos tradicionales los estudiantes pueden responder de manera correcta simplemente memorizando la explicación del profesor, sin la necesidad de entender los conceptos que se utilizan o los procesos implicados.

3.1. Análisis del aprendizaje de los estudiantes

Los nuevos enfoques sobre la enseñanza de las ciencias plantean analizar los procesos de construcción y aprendizaje de los conocimientos científicos en los alumnos, considerando tanto representaciones internas como las externas. Esto plantea el reto de construir nuevas metodologías para la enseñanza y su evaluación.

De esta manera, los instrumentos de evaluación, deben permitir analizar las representaciones externas que tienen los estudiantes sobre un tema en específico, tales representaciones contienen elementos cognitivos que reflejan los procesos y concepciones de los estudiantes, así como las relaciones entre ellas, en las cuales se observa la estructura o conexión conceptual generada por los estudiantes. Estas representaciones

reflejan las propiedades o cualidades que asimilaron los estudiantes.

Además con estas representaciones externas es posible identificar los elementos que se esperan que usen los estudiantes y así mismo identificar los elementos que no corresponden a los conceptos revisados y carecen de sentido.

3.1.1. La enseñanza de las ciencias

En el enfoque representacional se plantea que, los estudiantes puedan generar buenas representaciones, capaces de dar una explicación adecuada sobre el tema. Para esto es necesario que los estudiantes tengan acceso a diferentes representaciones de un mismo concepto; por ejemplo, en el estudio del concepto ADN, al estudiante se le puede presentar cómo luce gráficamente el ADN, después se le puede explicar de manera verbal su composición y por último una explicación escrita del mismo, de esta manera, el alumno tuvo contacto con tres representaciones distintas para generar una representación interna, con la cual pueda elaborar una explicación (representación externa) adecuada del concepto.

Una explicación adecuada debe mostrar los elementos conceptuales esperados, de acuerdo con los aspectos de los conceptos o fenómenos que se le han enseñado a los estudiantes [1]. Cabe señalar que en esta explicación muestra indicios de las ideas del estudiante expresadas con sus propios recursos, entonces estas representaciones pueden ser usadas para analizar el aprendizaje de los alumnos y la forma de enseñanza de los profesores.

En [7] citado en [1], plantea cinco niveles de competencia para la explicitación representacional:

- **Nivel 1**, representación como elemento figural.
- **Nivel 2**, nivel básico o primario de habilidad simbólica.
- **Nivel 3**, uso sintáctico de representaciones formales.
- **Nivel 4**, uso semántico de representaciones formales.
- **Nivel 5**, uso reflexivo y retorico de las representaciones.

Con estos niveles se puede dar un valor a las representaciones externas de los alumnos.

3.1.2. Construcción de un cuestionario para el análisis del aprendizaje de los estudiantes

Para construir un instrumento que sea capaz de analizar el aprendizaje de los estudiantes, debe hacer explícitas las representaciones de los estudiantes, para esto se considera el uso de preguntas que permitan hacer uso de los niveles de competencia para la explicación representacional.

Las preguntas que usan los niveles de competencia hacen que el estudiante haga uso de sus representaciones externas, para elaborar una explicación a una situación y sus respuestas muestran la estructura y la comprensión de los conceptos en diferentes situaciones.

En [1], se plantea que las preguntas usadas deben cumplir ciertos aspectos. Estos aspectos son:

- "Atender a conocimientos que, en principio, han sido analizados a lo largo de las trayectorias escolares de los estudiantes, en otras palabras, deben usar temas que fueron impartidos en los ciclos escolares ..."
- "Presentar situaciones cotidianas fácilmente interpretables por los estudiantes ..."
- "Susceptibles de ser representadas de diversas formas ..."
- "Posibilitar un proceso de reelaboración de explicaciones a lo largo del instrumento ..." ([1], página 6).

Con estos aspectos se caracterizan los elementos necesarios para realizar un análisis de las representaciones, porque con estas representaciones se puede determinar los patrones en las respuestas que reflejan las ideas o conceptos con las que cuentan los estudiantes y las diferencias entre estudiantes.

3.1.3. Metodología del análisis tradicional

Siguiendo el planteamiento anterior, dentro del proyecto de investigación "Procesos de transformación de las representaciones científicas en los estudiantes del bachillerato bajo un entorno multi representacional apoyado con tecnologías digitales", se creó como instrumento un cuestionario con preguntas de respuesta abierta, para que los estudiantes de bachillerato fueran estimulados a responder de diversas formas usando sus representaciones externas, los estudiantes responderán de forma escrita y elaborando dibujos según lo requiera cada una de las preguntas.

El desarrollo del cuestionario se fundamenta en el enfoque de la construcción de las representaciones externas como herramientas cognitivas. Permitiendo a los estudiantes organizar y exponer sus ideas de una temática particular.

Se eligió el tema de genética porque permite a los estudiantes de bachillerato, usar conocimiento básicos de genética para explicar las variaciones y mutaciones o alteraciones, con el fin de conocer si los estudiantes tienen la posibilidad de reconocer que se hereda, quién y qué hereda, y cómo lo hereda. El proceso de construcción que se siguió en el cuestionario antes mencionado consta de tres etapas:

- *La fase de construcción del instrumento*, en la cual participaron 10 alumnos de la Facultad de Química, 10 alumnos de bachillerato y 3 especialistas en Biología, los cuales construyeron el cuestionario.
- *La fase de validación del instrumento*, en la cual participaron 3 especialistas en Biología, donde se cubrieron aspectos como que las respuestas fueran lo más detalladas y amplias dentro de lo posible, así mismo, se analizó el nivel de complejidad de las respuestas esperadas y las representaciones externas usadas.
- *La fase de validación estadística y calificación*, se aplicó el cuestionario a 387 estudiantes que cursan el sexto año de bachillerato en la Escuela Nacional Preparatoria (ENP) y el quinto año en el Colegio de Ciencias y Humanidades (CCH). Para calificar los cuestionarios se construyó una rúbrica, con el fin de tener una asignación consistente del nivel de competencia en las respuestas de los estudiantes; además se realizó una validación adicional, para comprobar que el nivel asignado a cada respuesta sea el correcto siguiendo las reglas de la rúbrica. La rúbrica se basa en el trabajo presentado por Kozma y Russell [1].

3.1.4. Desventajas de la análisis tradicional

En la metodología actual del proceso de calificación de las respuestas existe una desventaja, en el cual los investigadores apoyados con algunos estudiantes, calificaron todas las respuestas lo que les llevó varios meses, ya que tuvieron que leer cada una de las preguntas y asignarle un nivel de competencia según indica la rúbrica.

3.2. Proceso de evaluación asistida

En esta tesis se propone el uso de técnicas de minería de texto para apoyar a los investigadores en el proceso de asignación del nivel de competencia en un menor tiempo. Para ello se construirá un modelo capaz de clasificar o calificar automáticamente las respuestas de los estudiantes. Este modelo es entrenado con una muestra de las respuestas, esta muestra ya debería contar con el nivel de competencia asignado, es decir, que los investigadores deben calificarlas previamente.

Una vez entrenado, el modelo es capaz de calificar las respuestas que aún no cuentan con su nivel de competencia. Este modelo puede ser personalizado en las etapas generales de la minería de texto para que se ajuste a las necesidades del investigador, y esta personalización se realiza por medio de una aplicación web que permite:

- Cargar la muestra de entrenamiento.
- Cargar una lista personalizada de stopwords.
- Cargar una lista de términos equivalentes.
- Visualizar la distribución de términos más frecuentes, n-gramas y gráfica de conceptos.
- Entrenar y probar el modelo.
- Usar el modelo para calificar automáticamente.
- Refinamiento del modelo.

A esta personalización de las etapas generales la denominamos proceso de evaluación asistida. En el cuadro 3.1 se observa la equivalencia entre las etapas de este proceso con las etapas generales de la minería de textos mencionadas en el capítulo anterior.

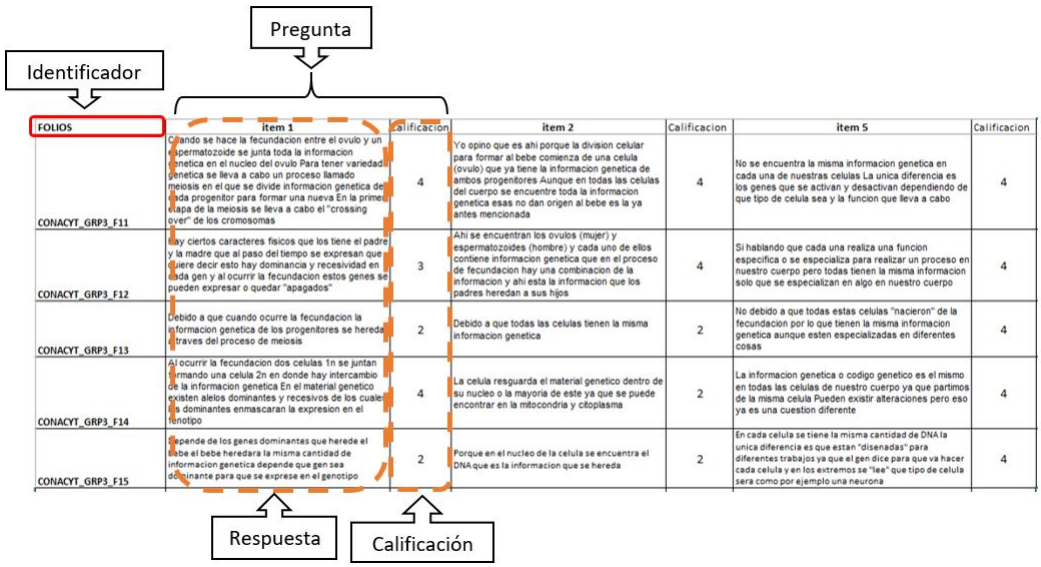
A continuación se describen con detalle las etapas del proceso de evaluación asistida.

3.2.1. Carga de la muestra de entrenamiento

Para iniciar con el proceso de evaluación asistida, se necesita que el cuestionario esté en un archivo que pueda manejar la aplicación web. Para esto, el cuestionario se debe capturar en una hoja de cálculo, en otras palabras, el cuestionario está en un archivo de tipo XLSX (Hoja de cálculo). Este archivo tiene la siguiente estructura: en la primera columna se indica un folio para identificar la respuesta de cada estudiante, además por cada pregunta del cuestionario hay un par de columnas, una para el texto de las respuestas y la otra con el nivel de competencia asignado por el evaluador. Un ejemplo de este cuestionario se muestra en la figura 3.1.

Etapas del proceso de evaluación asistida	**Etapas generales de la minería de texto**
Cargar la muestra de entrenamiento. Cargar una lista personalizada de stopwords. Cargar una lista de términos equivalentes.	Tareas de preprocesamiento.
Entrenar y probar el modelo. Usar el modelo para calificar automáticamente.	Aplicación de algoritmos de minería de texto.
Visualización de la distribución de términos, n-gramas y gráfica de conceptos.	Visualización de los resultados.
Refinamiento del modelo.	Refinamiento del proceso.

Cuadro 3.1: Muestra la equivalencia de las etapas del proceso de evaluación asistida y minería de texto.



FOLIOS	item 1	Calificacion	item 2	Calificacion	item 5	Calificacion
CONACYT_GRP3_F11	Cuando se hace la fecundacion entre el ovulo y un espermatozoide se junta toda la informacion genetica en el nucleo del ovulo Para tener variedad genetica se lleva a cabo un proceso llamado meiosis en el que se divide informacion genetica de cada progenitor para formar una nueva En la primera etapa de la meiosis se lleva a cabo el "crossing over" de los cromosomas	4	Yo opino que es ahi porque la division celular para formar al bebe comienza de una celula (ovulo) que ya tiene la informacion genetica de ambos progenitores Aunque en todas las celulas del cuerpo se encuentre toda la informacion genetica esas no dan origen al bebe es la ya antes mencionada	4	No se encuentra la misma informacion genetica en cada una de nuestras celulas La unica diferencia es los genes que se activan y desactivan dependiendo de que tipo de celula sea y la funcion que lleva a cabo	4
CONACYT_GRP3_F12	Hay ciertos caracteres fisicos que los tiene el padre y la madre que al paso del tiempo se expresan que quiere decir esto hay dominancia y recesividad en cada gen y al ocurrir la fecundacion estos genes se pueden expresar o quedar "apagados"	3	Ahi se encuentran los ovulos (mujer) y espermatozoides (hombre) y cada uno de ellos contiene informacion genetica que en el proceso de fecundacion hay una combinacion de la informacion y ahi esta la informacion que los padres heredan a sus hijos	4	Si hablando que cada una realiza una funcion especifica o se especializa para realizar un proceso en nuestro cuerpo pero todas tienen la misma informacion solo que se especializan en algo en nuestro cuerpo	4
CONACYT_GRP3_F13	Debido a que cuando ocurre la fecundacion la informacion genetica de los progenitores se hereda a traves del proceso de meiosis	2	Debido a que todas las celulas tienen la misma informacion genetica	2	No debido a que todas estas celulas "nacieron" de la fecundacion por lo que tienen la misma informacion genetica aunque esten especializadas en diferentes cosas	4
CONACYT_GRP3_F14	Al ocurrir la fecundacion dos celulas 1n se juntan formando una celula 2n en donde hay intercambio de la informacion genetica En el material genetico existen alelos dominantes y recesivos de los cuales los dominantes enmascaran la expresion en el fenotipo	4	La celula resguarda el material genetico dentro de su nucleo o la mayoria de este ya que se puede encontrar en la mitocondria y citoplasma	2	La informacion genetica o codigo genetico es el mismo en todas las celulas de nuestro cuerpo ya que partimos de la misma celula Pueden existir alteraciones pero eso ya es una cuestion diferente	4
CONACYT_GRP3_F15	Depende de los genes dominantes que herede el bebe el bebe heredara la misma cantidad de informacion genetica depende que gen sea dominante para que se exprese en el genotipo	2	Porque en el nucleo de la celula se encuentra el DNA que es la informacion que se hereda	2	En cada celula se tiene la misma cantidad de DNA la unica diferencia es que estan "disenadas" para diferentes trabajos ya que el gen dice para que va hacer cada celula y en los extremos se "lee" que tipo de celula sera como por ejemplo una neurona	4

Figura 3.1: Fragmento del cuestionario capturado por los investigadores.

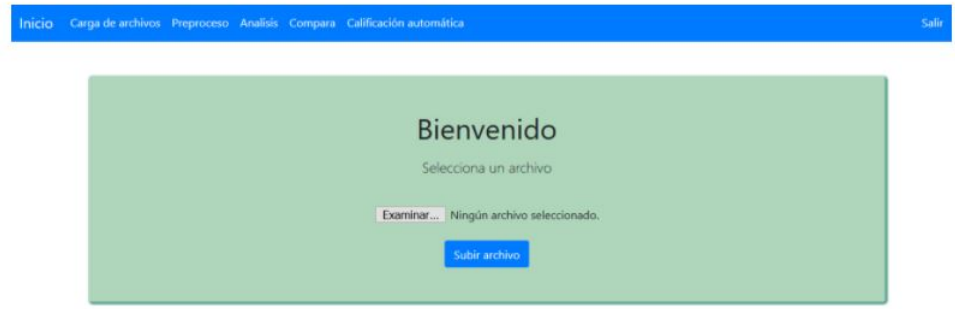


Figura 3.2: Muestra la pantalla para cargar el cuestionario.

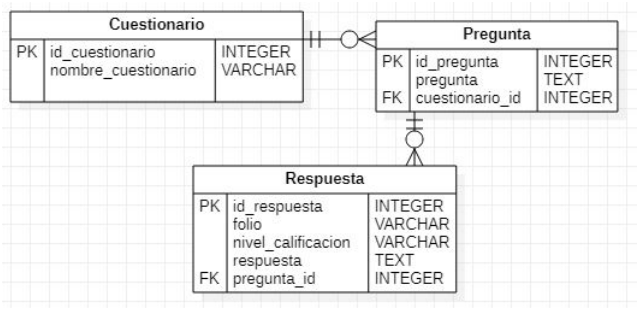


Figura 3.3: Diagrama de la base de datos que almacena a las preguntas y sus respuestas del cuestionario.

La primera etapa del proceso de evaluación asistida es cargar la muestra de entrenamiento que es el archivo antes descrito (la hoja de cálculo), y para cargar el archivo se utiliza la pantalla que se muestra en la figura 3.2, en la cual se puede elegir y cargar el archivo en la aplicación web.

Cuando se carga el archivo, la aplicación web almacena la información en una base de datos para tener un manejo fácil y sencillo de las respuestas cargadas. Previo a este almacenamiento, la aplicación web remueve los caracteres especiales que hay dentro del texto de las respuestas cargadas, los cuales pueden causar problemas de codificación. Por ejemplo, una comilla (’ o ”) que puede ser interpretada como una cadena incompleta.

El diagrama de la base de datos que se muestra en la figura 3.3, representa la estructura que sigue la base de datos para almacenar el cuestionario cargado. Este diagrama está compuesto por tres tablas, la primera tabla es “Cuestionario” que tiene su identificador y el nombre del cuestionario; la segunda tabla es “Pregunta”, la cual almacena las preguntas con un identificador, el texto de la pregunta y el identificador del cuestionario al que pertenece. La tercera tabla es “Respuesta”, esta tabla contiene para cada respuesta, un identificador, el folio que se le asignó previamente, su nivel de competencia, el texto que representa la respuesta y el identificador de la pregunta asociada a la respuesta.

Cuando el cuestionario es guardado correctamente en la base de datos, se muestra

Figura 3.4: Muestra el mensaje después de cargar el cuestionario.

```
el
la
lo
las
los
su
aqui
mio
tuyo
ellos
ellas
nos
nosotros
```

Figura 3.5: Fragmento del archivo que contiene una lista de stopwords.

un mensaje que lo notifica (figura 3.4) e indica el siguiente paso que es “Preproceso”, en la cual se lleva a cabo la carga de una lista personalizada de stopwords y de términos equivalentes. En el caso de que el cuestionario no respete el formato o contenga columnas inválidas, se mostrará un mensaje de error y no se guardará en la base de datos.

3.2.2. Carga de una lista personalizada de stopwords

Los estudiantes usan las representaciones externas para expresar las respuestas de las preguntas del cuestionario, por esto es importante identificar a las palabras o términos que pueden estar asociados a las representaciones externas, y se propone eliminar las palabras que no son necesarias para este análisis. Estas palabras pueden ser tratadas como stopwords o palabras irrelevantes que se mencionaron en el capítulo anterior, porque pueden ser eliminadas sin perder información relevante. Al eliminar estas palabras se reduce la cantidad de los términos usados en las respuestas, y con esta reducción es más sencillo identificar a las representaciones externas.

La lista de stopwords puede ser cargada en la aplicación web usando un archivo de tipo de texto plano, tradicionalmente .txt. Este archivo tiene en cada renglón una palabra o término que componen la lista de stopwords, en la figura 3.5 se presenta un ejemplo de la lista que contiene algunos stopwords comunes.

El archivo que contiene la lista de stopwords es cargado a la aplicación web, usando la pantalla de carga (figura 3.6) para la lista de stopwords.

Al cargar la lista de stopwords en la aplicación web, también es almacenada en una base de datos, para tener una recuperación rápida de ella. En la figura 3.7 se presenta la tabla del diagrama de la base de datos que modela la lista de stopwords, la cual está compuesta por su identificador de la lista, la lista de stopwords en una cadena de texto

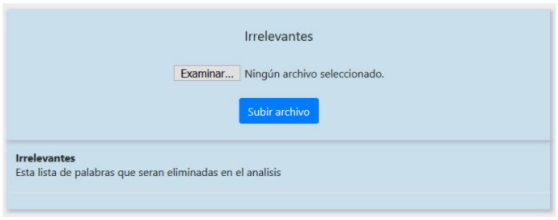


Figura 3.6: Pantalla de carga para la lista de stopwords.

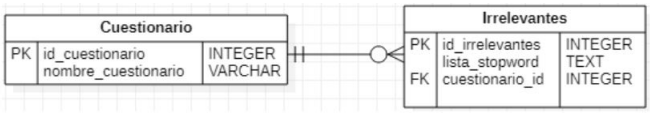


Figura 3.7: Diagrama para la lista de stopwords.

y el identificador del cuestionario al que está asociada la lista. Esto permite almacenar distintas listas de stopwords para distintos cuestionarios.

Al almacenar la lista de stopwords en la base de datos, la aplicación web muestra en pantalla las palabras que componen esta lista de stopwords personalizada (figura 3.8).

3.2.3. Carga de una lista de términos equivalentes

Durante el análisis tradicional se encontró que una misma representación externa puede ser referida de distintas maneras en las repuestas del cuestionario, lo que implica que varios términos pueden ser asociados a una misma representación externa. Para indicar los términos asociados a una misma representación externa, se plantea la carga de una lista de términos equivalentes, para sustituir estos términos por uno solo dentro de las respuestas.

Al sustituir varios términos por uno solo, se tiene un solo término para cada una de las representaciones externas en las respuestas, lo que reduce la diversidad de los términos sin pérdida de información.

Para cargar la lista de términos equivalentes en la aplicación web, es necesario que la lista esté en un archivo de tipo de texto plano, tradicionalmente en .txt. Este archivo tiene en cada uno de sus renglones una serie de términos (separados por una coma) que pueden ser sustituidos por el primero de la serie. Por ejemplo, ”padres, padre madre”, cada vez que aparezca “padre madre” dentro de una respuesta será sustituido por “padres”. En la figura 3.9 se muestra una lista de términos equivalentes.

En la figura 3.10 se muestra la pantalla de carga para la lista de términos equivalentes.

Cuando este archivo es cargado en la aplicación web, se almacena en la base de datos y se sustituyen los términos que hay dentro de los textos de las respuestas. El diagrama que se muestra en la figura 3.11 representa a la lista de términos equivalentes, el diagrama está compuesto por la tabla “Equivalencias” con el identificador de la serie de equivalencia, la serie de términos equivalentes y el identificador que indica al cuestionario pertenece. Y por la tabla de “Cuestionario” antes descrita.

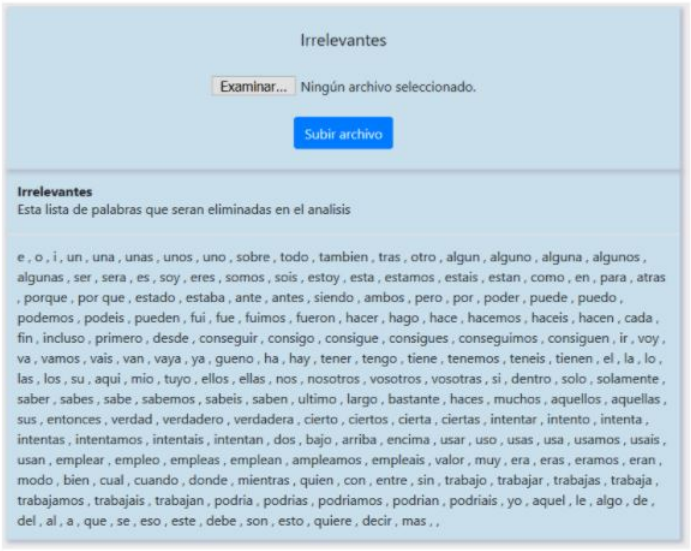


Figura 3.8: Muestra las palabras que componen a la lista de stopwords.

```
profase,meiosis
cruza,mezcla,combinacion,recombinacion
dna,and
variabilidad,variacion
genes,alelo,alelos,gen
dominantes,dominancia,dominates
mitad,parte,50%
gameto,gametos,ovulo,espermatozoide,esperma
recesivos,recesividad
entrecruzamiento,crossing over
padres,padre madre,progenitores,madre padre
cromosomas sexuales,xx,yy
dominantes recesivos,recesivos dominantes
genes dominantes recesivos,dominantes recesivos genes,genes recesivos dominantes
```

Figura 3.9: Archivo que contiene una lista de términos equivalentes.

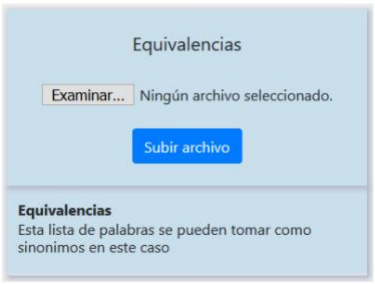


Figura 3.10: Pantalla de carga para la lista de términos equivalentes.

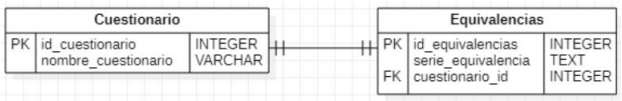


Figura 3.11: Diagrama para la lista de términos equivalentes.

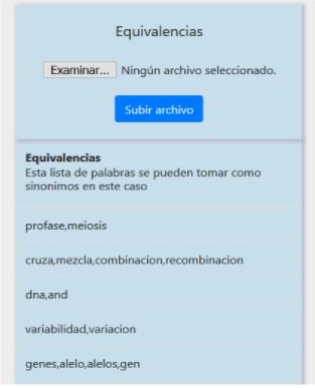


Figura 3.12: Muestra la lista de términos equivalentes.

Cuando la lista de términos equivalentes es almacenada en la base de datos, la aplicación web muestra la lista guardada en pantalla, en la figura 3.12 muestra una parte de la lista de términos equivalentes que se guardan en la base de datos.

Cabe señalar, que para el entrenamiento del modelo estas etapas son útiles, ya que estas reducen la cantidad de texto que hay que procesar y el aprendizaje del modelo le será más rápido y sencillo para aprender los patrones que le permita evaluar como un investigador lo hace.

3.2.4. Entrenar y probar el modelo

Durante el análisis tradicional, los investigadores pueden tardar en asignar el nivel de competencia a las respuestas del cuestionario, para ayudar a solucionar este problema, se plantea usar los algoritmos de minería de textos para reducir el tiempo de evaluación de las respuestas. Estos algoritmos son capaces de crear un modelo, el cual puede asignar un nivel de competencia a las respuestas en un tiempo menor.

Se decidió usar los algoritmos de minería de texto de tipo predictivo, en particular, los algoritmos de clasificación, porque estos algoritmos pueden realizar una asignación de una categoría a un texto o documento, para este caso, se utilizarán los algoritmos de clasificación para crear al modelo, el cual clasifiqué las respuestas usando los niveles de competencia antes mencionados.

Para que el algoritmo de minería pueda generar un modelo, necesita ser entrenado con una muestra del cuestionario ya calificado, en este entrenamiento el algoritmo aprende los patrones para asignar un nivel de competencia. La muestra es dividida en dos partes o conjuntos de acuerdo con el porcentaje indicado por el usuario, esto se muestra en la figura 3.13, la mayor parte es usada para entrenar al modelo y la otra parte para realizar las pruebas, estas pruebas sirven para verificar que el modelo tenga

Figura 3.13: Muestra la pantalla donde selecciona la pregunta y el porcentaje para entrenar al algoritmo.

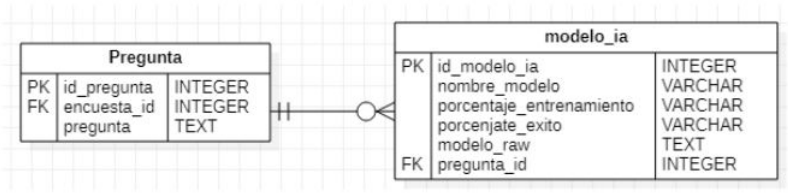


Figura 3.14: Muestra las tablas usadas para almacenar al modelo entrenado.

el comportamiento esperado.

Para comprobar que el comportamiento del modelo sea el esperado, se le pasa el conjunto prueba para que este les asigne un nivel de competencia, después se compara la asignación del modelo con la asignación original. El resultado de la prueba indica el total de respuestas que fueron asignadas correctamente.

En la aplicación web es posible entrenar un modelo (algoritmo de minería de texto), el cual será capaz de asignar un nivel de competencia a las respuestas. Cabe señalar que esto se realiza por cada una de las preguntas que contenga el cuestionario, es decir, se construye un modelo para cada pregunta.

Para entrenar al algoritmo es necesario elegir una pregunta del cuestionario y el porcentaje para dividir las respuestas en el conjunto de entrenamiento y de prueba. Por ejemplo en la figura 3.13 se muestra que se eligió la primera pregunta (Ítem 1), para entrenar al modelo el conjunto de entrenamiento es el 85 % del total de las respuestas cargadas y el 15 % restante se utiliza para realizar las pruebas,aunque en [2] se recomienda usar dos tercios de la muestra para entrenar al modelo y el tercio restante para evaluar al modelo. De esta manera la aplicación web crea un modelo para cada pregunta del cuestionario.

La muestra se elige de manera aleatoria para realizar el entrenamiento, las respuestas que fueron elegidas para el entrenamiento se almacena en la base de datos. Esta información se almacena en la tabla “Modelo_ia” del diagrama de la base de datos, esta tabla usa al identificador de la pregunta, también tiene el identificador del modelo creado, nombre del modelo, el porcentaje de las respuestas con el que fue entrenado, el porcentaje de éxito y el archivo del modelo generado. En la figura 3.14 se muestran las tablas “Respuestas” y “Modelo_ia” que representan el entrenamiento del algoritmo.

Una vez que la aplicación web finaliza la creación y el almacenamiento del modelo, la aplicación muestra los resultados obtenidos de la prueba, se muestra el porcentaje usado para entrenar al modelo, el porcentaje usado para las pruebas y el desempeño

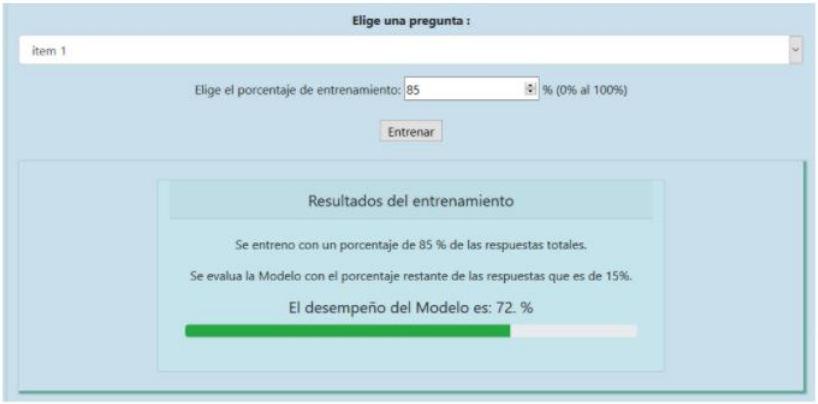


Figura 3.15: Muestra los resultados obtenidos después de entrenar al modelo.

del modelo. Por ejemplo, en la figura 3.15 se muestra los resultados obtenidos con las respuestas del Ítem 1, donde se tiene un entrenamiento con el 85 % de las respuestas, y el modelo obtuvo un 72 % de precisión para asignar el nivel de competencia correctamente.

Si se necesita repetir el entrenamiento del modelo para obtener un mayor porcentaje de precisión, solamente se tiene que repetir el proceso usando el botón “Entrenar”, hasta obtener un resultado adecuado para la calificación de las respuestas.

La parte principal de la propuesta, es el uso del modelo entrenado para asignar un nivel de competencia a las respuestas de la misma pregunta. Para usar el modelo solo es necesario cargar un archivo de tipo XLSX (Hoja de cálculo), este archivo cuenta con la siguiente estructura: en la primera columna se indica un folio para identificar la respuesta de cada estudiante y otra columna para indicar las respuestas de la pregunta correspondiente al modelo. Un ejemplo de este archivo se muestra en la figura 3.16.

Este archivo puede ser cargado a la aplicación web usando la pantalla que se muestra en la figura 3.17 , en la cual se elige el archivo que tiene las repuestas sin nivel de competencia.

Al cargar las respuestas en la aplicación web, estas son almacenadas en la base de datos, usando la tabla “Evalua_respuestas” que se muestra en la figura 3.18, la cual tiene el identificador de la respuesta, el identificador del modelo usado, el texto de la respuesta, el folio asignado y nivel dado por el modelo. A continuación el modelo les asigna un nivel de competencia a estas respuestas y lo almacena en la base de datos.

Desde la aplicación se pueden recuperar las respuestas con la asignación de competencia, así como la muestra usada para el entrenamiento. En la figura 3.19 se muestra el menú de descarga, el cual contiene dos opciones que son:

- Cuestionario base: descarga un archivo de tipo XLSX, el cual contiene la muestra completa que fue cargada en la aplicación web.

- Cuestionario calificado: descarga un archivo de tipo XLSX, con la muestra y el nivel de competencia asignado por el modelo creado.

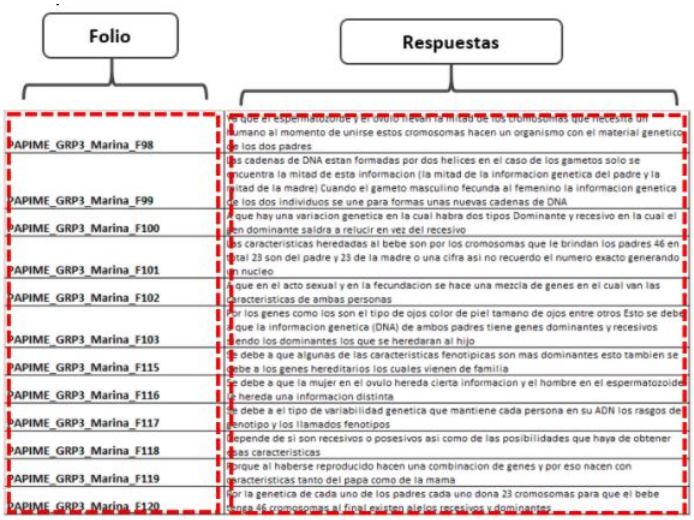


Figura 3.16: Fragmento del cuestionario con respuestas sin nivel de competencia.

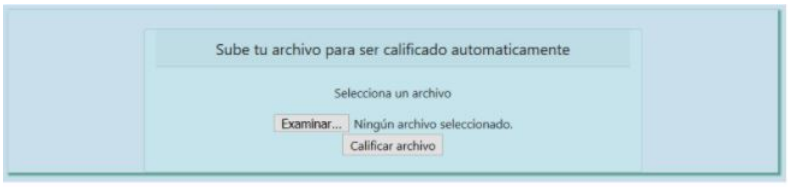


Figura 3.17: Muestra la pantalla de carga para las respuestas sin nivel de competencia.

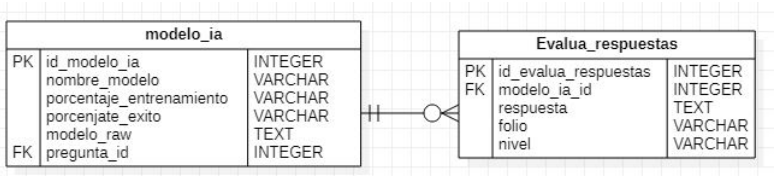


Figura 3.18: Muestra el diagrama de las respuestas evaluadas por el modelo.

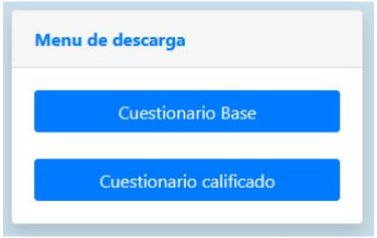


Figura 3.19: Muestra el menú de descarga para cuestionario base y el calificado por el modelo.

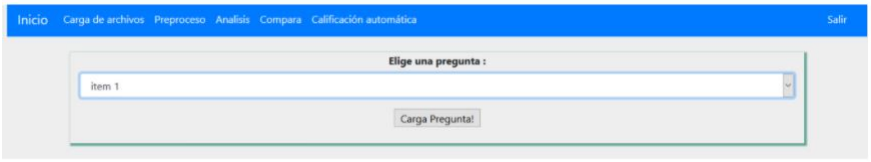


Figura 3.20: Muestra la elección de la pregunta para la visualización de información.

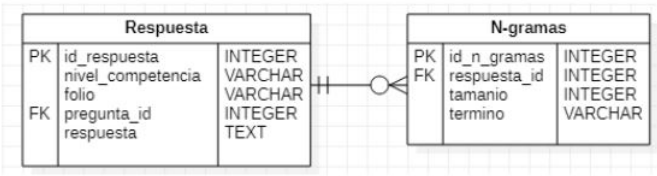


Figura 3.21: Muestra las tablas del diagrama de la base de datos usadas para guardar los n-gramas.

3.2.5. Visualización de la distribución de términos, n-gramas

En el análisis tradicional los investigadores buscan los términos más comunes en las preguntas, para identificar las representaciones externas y así entender cómo los estudiantes usan esas representaciones externas para expresar su entendimiento del tema.

En la etapa de visualización se muestra la distribución de términos, n-gramas y gráfica de conceptos de las respuestas. El objetivo de visualizar esta información es identificar las representaciones externas que hay en las respuestas.

La aplicación web presenta los n-gramas (de 1, 2 y 3 términos) más usados y gráficas conceptuales. Un n-grama es la serie consecutiva de n palabras dentro de una oración o texto. Por ejemplo si tenemos la oración “una parte de la informacián genética”, los n-gramas de 1 término serían “una”, “parte”, “de”, “la”, “información”, “genética”.

Las gráficas conceptuales son redes de palabras, estas redes son construidas por la aplicación web usando los diez trigramas más frecuentes de las respuestas de una pregunta, las gráficas conceptuales sirven para observar y explorar las relaciones que tienen los términos dentro de las respuestas, así mismo se puede observar la relación que hay entre las gráficas conceptuales y las respuestas.

Para que la aplicación web pueda mostrar la tabla de términos y la gráfica conceptual, hay que elegir una pregunta del cuestionario cargado como se muestra en la figura 3.20, donde se elige la primera pregunta (Ítem 1).

Cuando se tiene una pregunta seleccionada, la aplicación web crea los n-gramas de las respuestas de la pregunta seleccionada y son almacenados en la base de datos. En la figura 3.21 se muestra el diagrama de las tablas “Respuesta” y “N-gramas”, la tabla “N-gramas” tiene el identificador del n-grama, el identificador de la respuesta al que pertenece, el tamaño del n-grama, en este caso (1, 2, 3) y el término correspondiente.

Al terminar de calcular y almacenar los n-gramas la aplicación web muestra una tabla de términos y la gráfica de conceptos para cada uno de los niveles de competencia que tiene la pregunta, y una tabla de términos y una gráfica de todas las respuestas de la pregunta. Con esta presentación de las respuestas, los investigadores pueden identificar

Se muestra la tabla y el arbol : General

Palabra	Frecuencia	Bigrama	Frecuencia	Trigrama	Frecuencia
caracteristicas	360	informacion genetica	224	parte informacion genetica	21
informacion	358	padre madre	132	50% informacion genetica	20
genetica	332	material genetico	68	mitad informacion genetica	19
padre	320	parte informacion	35	informacion genetica bebe	16
madre	306	gen dominante	34	tanto madre padre	14
bebe	262	23 cromosomas	30	tanto padre madre	13
alelo	221	mitad informacion	29	informacion genetica padre	10
genes	201	46 cromosomas	28	aportan informacion genetica	10
cromosomas	158	madre padre	27	genetica padre madre	9
dominancia	129	50% informacion	26	expresion determinadas caracteristicas	8

Figura 3.22: Se muestran los diez n-gramas más frecuentes de la una pregunta, con $n \in \{1, 2, 3\}$.

las representaciones externas usadas en cada uno de los niveles de competencia.

La imagen que se muestra en la figura 3.22, es el listado de los diez n-gramas más frecuentes de una pregunta del cuestionario, además que cada n-grama tiene su frecuencia asociada.

Las gráficas conceptuales tienen la intención de visualizar los trigramas más frecuentes y mostrar las relaciones que tienen los trigramas, para esto la aplicación web crea una red de palabras, estas palabras son parte de los trigramas mostrados en la tabla de términos. Cada nodo que está en la red, está conectado con una arista dirigida, las cuales indican la forma en que se lee el trigrama, lo que permite reconstruir los trigramas usados. En la figura 3.23 se muestra el trigrama "expresión determinadas características", para identificar el inicio de un trigrama en la red, es ubicando al primer nodo que es pintado del color azul y el resto de color amarillo.

A partir de los términos en común de los trigramas, se obtiene una gráfica que

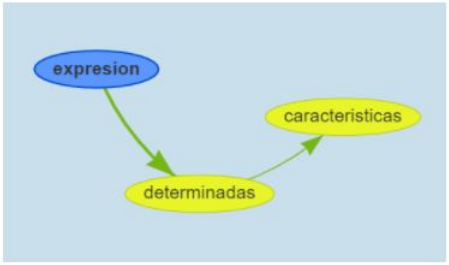


Figura 3.23: Muestra la representación del trigrama "expresión determinadas características" en la gráfica conceptual.

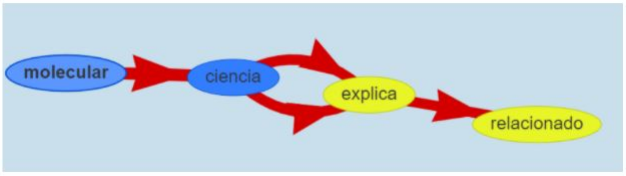


Figura 3.24: Muestra la conexión entre dos trigramas que comparten términos en común.

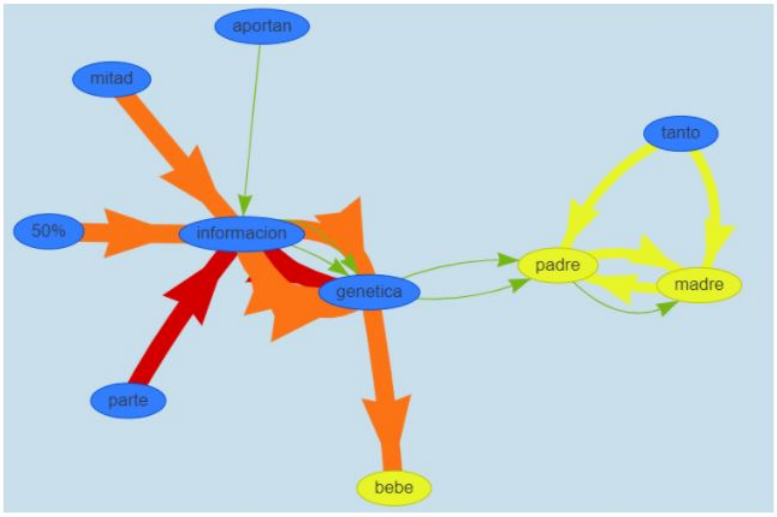


Figura 3.25: Muestra la relación de color y tamaño que hay en las flechas que unen a los nodos, dependiendo su frecuencia.

conecta los diferentes trigramas, en la figura 3.24 se muestran dos trigramas “ciencia explica relacionado” y “molecular ciencia explica”, estos dos trigramas comparten las palabras “ciencia” y “explica”, se comienza en “molecular” y “ciencia”, las cuales son pintadas de color azul porque ambas son el inicio de sus trigramas correspondientes, el resto de los nodos son pintados de color amarillo.

Las aristas de los nodos varían de tamaño (grosor) y de color, estos dependen de la frecuencia del trigrama, los trigramas con mayor frecuencia tiene un color más intenso y un mayor tamaño. La gama de intensidad de los colores son de rojo, anaranjado, amarillo y verde, en ese orden.

Por ejemplo, en la figura 3.25 se muestra el trigrama “parte información genética”, que tiene la mayor frecuencia en la tabla de la figura 3.22, las aristas que conectan a este trigrama son de color rojo y son las de mayor tamaño, a diferencia del trigrama “genética padre madre” que es de los que tienen una frecuencia menor, lo que implica que sus flechas son de color verde y son las más delgadas.

Para conocer la frecuencia a la que pertenecen los colores de las aristas, se puede usar la tabla de frecuencias por colores que se muestra en la misma pantalla, la cual

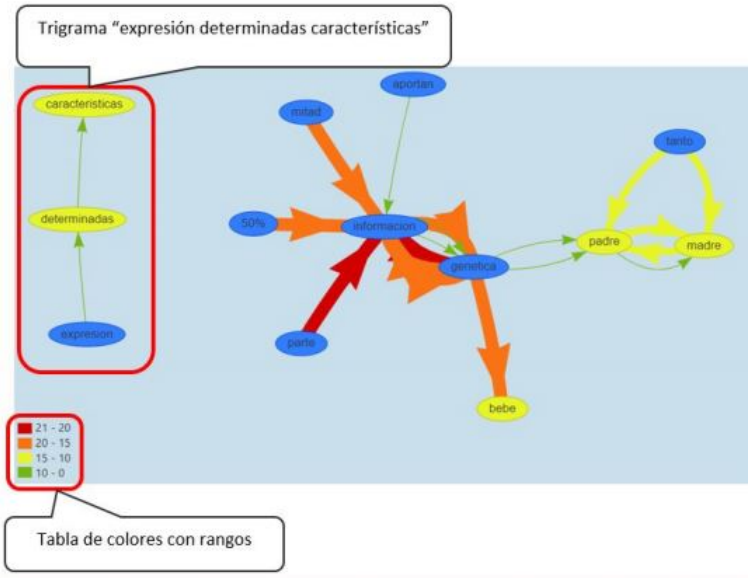


Figura 3.26: Muestra la relación de color y frecuencia que hay en las flechas que unen a los nodos.

indica el rango que representan cada uno de los colores usados. Cabe señalar que esta tabla se ajusta en función de la gama de los colores usados.

Continuando con el ejemplo de la tabla 3.22, se muestra en la figura 3.26 la tabla de colores con sus rangos usados, las frecuencias de todos trigramas de la tabla en la figura 3.26 van de 8 a 21, y se les asignan de manera dinámica los cuatro colores que hay disponibles para las aristas. Para conocer el rango en el que se encuentra un color, solo basta buscarlo en tabla de colores, como el trigrama "parte información genética" tiene flechas rojas, entonces su rango de frecuencia es 20 a 21 y el trigrama "genética padre madre" tiene color verde entonces su rango de frecuencia es 0 a 10.

Con estas gráficas los investigadores pueden ver cómo se enlazan los trigramas usados, y con ello pueden identificar las representaciones externas más relevantes que son usadas en las respuestas y cómo se enlazan.

3.2.6. Refinamiento del modelo

Esta etapa de refinamiento del modelo se repiten algunas etapas previas del proceso de evaluación asistida, esto con el fin de obtener mejores resultados. Por ejemplo, se puede agregar palabras a la lista de términos irrelevantes y observar los cambios. Se recomienda repetir los pasos del proceso, es decir, cargar la muestra de entrenamiento para tener el cuestionario original sin ningún cambio previo, después cargar la nueva lista de stopwords y términos equivalentes, continuando con el entrenamiento, prueba del modelo, y finalmente analizar la visualización de las representaciones externas.

Resumen

En resumen, dentro del proyecto "Procesos de transformación de las representaciones científicas en los estudiantes del bachillerato bajo un entorno multi representacional apoyado con tecnologías digitales" se investigaron la formas de la enseñanza de las

ciencias en estudiantes de bachillerato.

Se propuso observar el uso de las representaciones externas por parte de los estudiantes, a través de un cuestionario. Este cuestionario fue elaborado para que los estudiantes proporcionen una explicación sobre el tema de genética .

El cuestionario es evaluado con un nivel de competencia para cada una de las respuestas, el cual establece el nivel de dominio que tiene el estudiante sobre el tema.

La asignación del nivel de competencia de los cuestionario, toma bastante tiempo. Para apoyar a esta actividad se propuso usar las técnicas de minería de textos, para realizar una asignación del nivel de competencia a las respuestas de manera automática.

En este capítulo se describió la metodología del proceso de evaluación asistida que ayuda a los investigadores en esta tarea, así como la aplicación web que apoya la metodología.

Capítulo 4

Experimentos y resultados

En este capítulo se muestran los resultados obtenidos al aplicar la metodología "proceso de evaluación asistida", la cual tiene la función de dar un apoyo a los investigadores en el proceso de asignación del nivel de competencia para un cuestionario. El cuestionario utilizado se diseñó como parte del proyecto de investigación llamado "Procesos de transformación de las representaciones científicas en los estudiantes del bachillerato bajo un entorno multi representacional apoyado con tecnologías digitales", el cual analiza la enseñanza de las ciencias en el temática de genética.

En el capítulo 2 se mencionó que existen dos tipos de algoritmos de minería de texto: descriptivos y predictivos. Los algoritmos descriptivos agrupan a los datos según las características que tienen en común. Los algoritmos predictivos realizan descubrimientos de patrones para hacer predicciones en los datos, estos algoritmos se pueden usar para la clasificación de datos. La metodología propuesta está enfocada en asignar los niveles de competencia a las repuestas del cuestionario, para obtener esta evaluación se utiliza un algoritmo de tipo predictivo.

4.1. Pruebas entre diferentes algoritmos de clasificación

En el capítulo 2, se mencionaron algunos algoritmos de minería de texto de tipo predictivo que se consideraron en este trabajo, los cuales son los árboles de decisión, clasificadores estadísticos (naive bayes), redes neuronales y máquinas de soporte vectorial. En esta sección se analiza el desempeño de estos algoritmos para elegir el que sea mejor para resolver la asignación de niveles de competencia. Para esto se consideró la implementación de estos algoritmos que se incluyen en el software Weka [8].

Desde Weka es posible realizar comparaciones entre la precisión de varios algoritmos de minería de textos, de una manera sencilla y práctica. En las pruebas realizadas, se utilizaron las respuestas de la primera pregunta del cuestionario antes mencionado con sus seis niveles de competencia asignados previamente. Esta pregunta contiene un total de 387 respuestas, en la figura 4.1 se muestra la distribución de estas respuestas dentro de los niveles de competencia.

Para utilizar los algoritmos que tiene Weka, se necesita que el cuestionario esté en un archivo que pueda manejar Weka. El cuestionario original está en formato de una hoja de cálculo de Excel (XLS) y se transformó a un archivo con formato nativo de Weka con extensión arff.

El formato arff de Weka está compuesto por tres partes: cabecera, declaración de

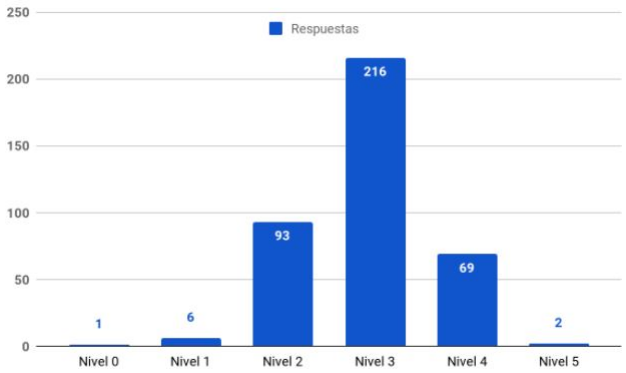


Figura 4.1: Muestra la asignación de las respuestas.

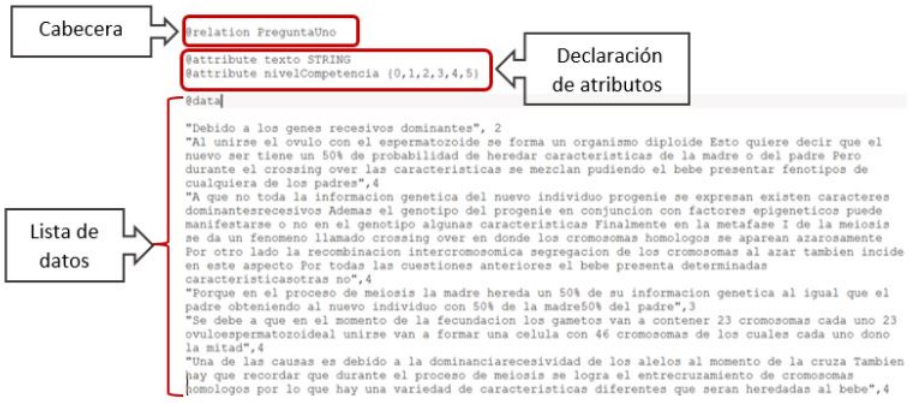


Figura 4.2: Fragmento del cuestionario en formato arff.

atributos y lista de datos [8]. Por ejemplo, en la figura 4.2 se muestra la primera pregunta del cuestionario en formato nativo de Weka. En la cabecera tiene la etiqueta @relation, esta lleva un nombre para identificar la relación que hay en el archivo. En la declaración de atributos tiene la etiqueta @attribute, esta es usada para indicar el tipo y nombre de cada atributo de la relación, en este caso una respuesta tiene un atributo de tipo texto (el texto de la respuesta) y un nivel de competencia (un valor en el conjunto $\{0, 1, 2, 3, 4, 5\}$). Finalmente en la lista de datos se indica cada una de las respuestas de la primera pregunta con sus niveles de competencia.

4.1.1. Análisis del rendimiento de los árboles de decisión.

Para analizar el rendimiento de los algoritmos de minería de texto, se realizaron cuatro particiones del total de respuestas para el entrenamiento y la prueba del modelo, además se realizaron 5 ejecuciones para cada partición. A continuación se enlistan las particiones:

Número de partición	Porcentaje de entrenamiento	Resultado de precisión
1	60 % (234 respuestas)	50 %
2	70 % (271 respuestas)	56 %
3	80 % (310 respuestas)	48 %
4	90 % (349 respuestas)	61 %

Cuadro 4.1: Muestra los resultados de las ejecuciones de prueba para los árboles de decisión.

- 60 % (234 respuestas) y 40 % (153 respuestas)
- 70 % (271 respuestas) y 30 % (116 respuestas)
- 80 % (310 respuestas) y 20 % (77 respuestas)
- 90 % (349 respuestas) y 10 % (38 respuestas)

Los árboles de decisión fue el primer algoritmo en esta prueba. En el Cuadro 4.1 se muestran los mejores resultados de las ejecuciones hechas para evaluar la precisión de los árboles de decisión.

En la figura 4.3 se muestra la evaluación del modelo entrenado con un 90 % de la muestra para entrenar al modelo y un 10 % para la evaluación del modelo. Estos resultados son representados en una matriz de confusión.

Con la matriz de confusión es posible observar las asignaciones correctas e incorrectas realizadas por el modelo entrenado. Así mismo, es posible observar el conteo de la asignación de los niveles de competencia. Además se tienen las gráficas conceptuales y el apartado con el cual se puede usar el modelo entrenado para evaluar una muestra del cuestionario que no tiene el nivel de competencia. Lo que permite decidir si el modelo tiene el comportamiento esperado para apoyar la metodología propuesta.

El total de las respuestas que fueron asignadas correctamente se encuentran en la diagonal de la matriz de confusión y las respuestas con asignación incorrecta se encuentran fuera de la diagonal, debajo de la columna a la que es asignada. Por ejemplo, para las respuestas con nivel de competencia igual a dos (c), el total de respuestas que son asignadas correctamente están en la entrada (c,c) y las que son asignadas incorrectamente se encuentran en la entrada (d,c), entonces se encontraron 3 respuestas con asignación correcta y 7 respuestas con asignación incorrecta de nivel de competencia igual a dos.

En la figura 4.3 se muestra el modelo entrenado con un 90 % de la muestra, el cual tiene un buen desempeño en los niveles tres (c) y cuatro (d), los cuales son dos de los tres niveles en los que se concentran la mayoría de las respuestas (como se puede ver en la figura 4.1). En los otros niveles de competencia hay un menor desempeño.

El comportamiento del modelo entrenado, también es analizado usando las diferentes particiones. En la figura 4.4 se muestra el modelo entrenado con un 80 % de las respuestas (izquierda) y el modelo entrenado con 70 % de las respuestas (derecha). Se observa que ambos modelos tienen un buen desempeño solo para el nivel tres.

4.1.2. Análisis del rendimiento de Naive bayes.

El segundo algoritmo de minería analizado es Naive bayes, en el cuadro 4.2 se muestra los resultados obtenidos de las evaluaciones del modelo, usando las diferentes muestras de entrenamiento antes descritas.

```
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  0  0  0  1  0  0 |  a = 0
  0  0  1  0  0  0 |  b = 1
  0  0  3  7  0  0 |  c = 2
  0  0  5 19  0  0 |  d = 3
  0  0  1  0  2  0 |  e = 4
  0  0  0  0  0  0 |  f = 5
```

Figura 4.3: Resultado de la evaluación del modelo entrenado.

Modelo entrenado con un 80%

```
a  b  c  d  e  f   <-- classified as
0  0  0  1  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  0  1 14  0  0 |  c = 2
0  0 14 30  2  0 |  d = 3
0  0  2  5  6  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Modelo entrenado con un 70%

```
a  b  c  d  e  f   <-- classified as
0  0  1  0  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  0  5 15  1  0 |  c = 2
0  0 16 50  6  0 |  d = 3
0  0  3  7 10  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Figura 4.4: Análisis del comportamiento del modelo entrenado con dos muestras diferentes del 80 % y 70 %.

Número de partición	Porcentaje de entrenamiento	Resultado de precisión
1	60 % (234 respuestas)	59 %
2	70 % (271 respuestas)	61 %
3	80 % (310 respuestas)	62 %
4	90 % (349 respuestas)	69 %

Cuadro 4.2: Muestra los resultados de las ejecuciones de prueba para Naive bayes.

```
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  1  0  0  0  0  0 |  a = 0
  0  0  1  0  0  0 |  b = 1
  0  0  4  6  0  0 |  c = 2
  0  0  3 20  1  0 |  d = 3
  0  0  0  1  2  0 |  e = 4
  0  0  0  0  0  0 |  f = 5
```

Figura 4.5: Resultado de la evaluación del modelo entrenado.

Modelo entrenado con un 80%

```
a  b  c  d  e  f   <-- classified as
1  0  0  0  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  0  8  7  0  0 |  c = 2
0  0  9 32  5  0 |  d = 3
0  0  1  5  7  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Modelo entrenado con un 70%

```
a  b  c  d  e  f   <-- classified as
1  0  0  0  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  1  9 11  0  0 |  c = 2
0  0 15 51  6  0 |  d = 3
0  0  1  9 10  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Figura 4.6: Comparación del comportamiento del modelo entrenado con dos muestras diferentes 80 % y 70 %.

En la figura 4.5 se muestra el modelo entrenado con el 90 % de las respuestas, la evaluación de esta muestra tiene un 69 % de precisión en la asignación del nivel de competencia, observando el comportamiento de este modelo en cada uno de los niveles, se tiene una buena asignación sólo en los niveles tres(d) y cuatro(e).

En la figura 4.6 se muestra el modelo entrenado con un 80 % y 70 % de las respuestas. En ambas muestras se tiene una buena asignación solo para el nivel tres.

4.1.3. Análisis del rendimiento de las redes neuronales.

Como se mencionó en el capítulo 2, las redes neuronales son conjuntos de neuronas conectadas entre sí, en estas pruebas de rendimiento se usaron varias redes neuronales para obtener el mayor porcentaje de precisión posible.

La estructura de las redes neuronales que usaremos, se describe indicando las capas internas y las neuronas que tiene cada una de estas capas, ya que la capa de entrada tiene una neurona por cada una de las distintas palabras de las respuestas y la capa de salida tiene una neurona por cada nivel de competencia. Por ejemplo, si la red es de 2 capas y de 5 neuronas, se tiene una red neuronal con una capa de entrada con 1266 neuronas que representa el texto de una respuesta, además de dos capas internas que tienen 5 neuronas en cada capa interna y en su capa de salida tiene 6 neuronas, en la figura 4.7 se muestran las capas internas y la de salida.

Weka cuenta con la opción de recomendar la cantidad de neuronas que puede tener una capa, esta opción usa el contenido de las respuestas para realizar dicha recomendación, algunas de las recomendaciones están basadas en el número de palabras distintas que hay en las respuestas (atributos = 1266) o los niveles de competencia (clases = 6) o simplemente el punto medio entre los atributos y las clases, esto corresponde a una

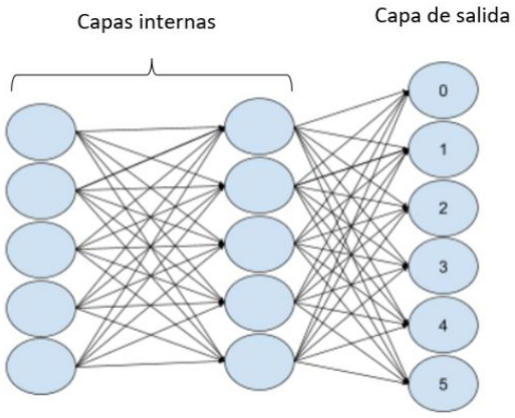


Figura 4.7: Red neuronal de 2 capas de 5 neuronas.

Red neuronal	Porcentaje
a= 636, 1 capa	61 %
o = 6, 1 capa	61 %

Cuadro 4.3: Resultado de las redes neuronales usando el 90 % de la muestra.

red neuronal de una capa de 636 neuronas ((atributos + clases) / 2) [8].

Se eligen la red neuronal del punto medio y los niveles de competencia para medir su desempeño con un entrenamiento del 90 % de las respuestas y un 10 % para la evaluación de la red. En el cuadro 4.3 se observa el desempeño de las redes neuronales propuestas, en ambas se tiene un 61 % de precisión.

Dado que en ambas redes neuronales solo tiene una capa interna, se prueba con dos redes neuronales diferentes, para verificar si es posible obtener un mayor porcentaje, una de estas redes es de 2 capas de 100 neuronas y la otra es de capas de 400, 100, 50 neuronas. En la primera se busca observar una red con más de una capa y con menos neuronas a comparación de las que tiene el punto medio y en la segunda observa una red con más capas y una cantidad similar a la del punto medio. En el cuadro 4.4 se muestra el resultado de ambas redes, donde obtiene un 61 % de precisión.

En la figura 4.8 se muestra las matrices de confusión de las redes neuronales de dos y tres capas antes mencionadas, en la red neuronal de tres capas se tienen predicciones más cercanas a su diagonal en comparación de la red de dos capas, por esta razón se elige la red neuronal de tres capas para realizar las pruebas de las distintas particiones de la muestra.

Red neuronal	Porcentaje
100 neuronas, 2 capa	61 %
400, 100, 50 neuronas, 3 capas	61 %

Cuadro 4.4: Resultado de las redes neuronales usando varias capas.

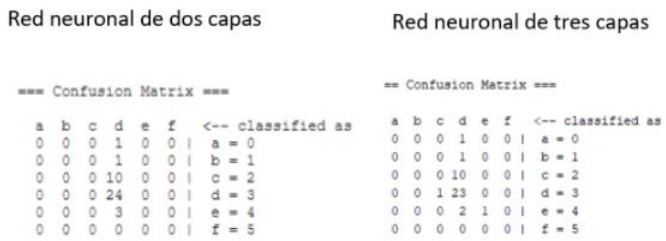


Figura 4.8: Comparación de matrices de confusión de dos redes neuronales.

Número de partición	Porcentaje de entrenamiento	Resultado de precisión
1	60 % (234 respuestas)	54 %
2	70 % (271 respuestas)	58 %
3	80 % (310 respuestas)	64 %
4	90 % (349 respuestas)	61 %

Cuadro 4.5: Muestra los resultados de las ejecuciones de prueba para las redes neuronales.

En el cuadro 4.5 se muestra los mejores resultados de las cuatro particiones realizadas para la red neuronal, el mejor resultado lo tiene la partición del 80 % de la muestra, la cual tiene un 64 % de precisión.

En la figura 4.9 se observa las matrices de confusión con la partición del 80 % y el 70 % de la muestra, se observa que el modelo entrenado con el 80 % tiene el mayor porcentaje de precisión.

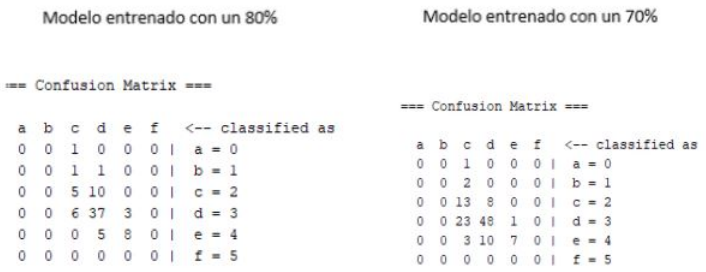
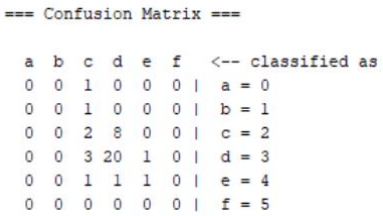


Figura 4.9: Comparación del comportamiento de la red neuronal entrenada con dos particiones diferentes.

Número de partición	Porcentaje de entrenamiento	Resultado de precisión
1	60 % (234 respuestas)	61 %
2	70 % (271 respuestas)	59 %
3	80 % (310 respuestas)	61 %
4	90 % (349 respuestas)	58 %

Cuadro 4.6: Muestra los resultados de las ejecuciones de prueba para las redes neuronales.



```
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  0  0  1  0  0  0 |  a = 0
  0  0  1  0  0  0 |  b = 1
  0  0  2  8  0  0 |  c = 2
  0  0  3 20  1  0 |  d = 3
  0  0  1  1  1  0 |  e = 4
  0  0  0  0  0  0 |  f = 5
```

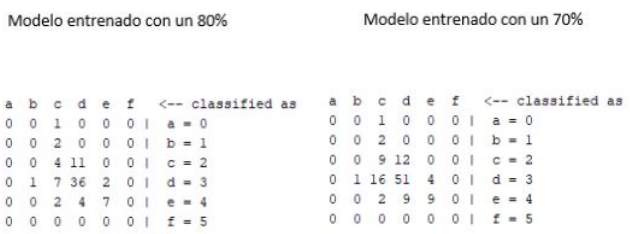
Figura 4.10: Modelo entrenado usando el 90 % de la muestra.

4.1.4. Análisis del rendimiento de las Máquinas de soporte vectorial.

El último algoritmo que se analiza son las máquinas de soporte vectorial (SMV), en el cuadro 4.6 se muestran los mejores resultados del entrenamiento para el modelo en cada una de sus particiones. En la mayoría de las particiones se obtuvo un porcentaje de precisión cercano al 60 %.

A continuación se muestra las diferentes matrices de confusión de las particiones usadas, la primera se muestra en la figura 4.10, la cual corresponde al 90 % de la muestra para entrenar al modelo y un 10 % para la evaluación con una precisión del 58 %.

En la figura 4.11 se muestra los modelos entrenados con un 80 % y 70 % de la muestra, el modelo que usa al 80 % de la muestra tiene el 61 % de precisión, que es la mayor precisión de las particiones propuestas para este algoritmo.



Modelo entrenado con un 80%

```
a  b  c  d  e  f   <-- classified as
0  0  1  0  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  0  4 11  0  0 |  c = 2
0  1  7 36  2  0 |  d = 3
0  0  2  4  7  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Modelo entrenado con un 70%

```
a  b  c  d  e  f   <-- classified as
0  0  1  0  0  0 |  a = 0
0  0  2  0  0  0 |  b = 1
0  0  9 12  0  0 |  c = 2
0  1 16 51  4  0 |  d = 3
0  0  2  9  9  0 |  e = 4
0  0  0  0  0  0 |  f = 5
```

Figura 4.11: Modelos entrenados usando el 80 % y 70 % de la muestra.

Algoritmo	Rango de éxito
Árboles de decisión	48 % al 61 %
Clasificadores estadísticos (Naive Bayes)	59 % al 69 %
Redes neuronales	54 % al 64 %
Máquinas de soporte vectorial	58 % al 61 %

Cuadro 4.7: Muestra el rango de los porcentajes de éxito, obtenido durante las pruebas en los algoritmos de minería de texto.

4.1.5. Comparación de los diferentes clasificadores

En el cuadro 4.7 se muestra el rango de precisión que se obtuvo de los algoritmos anteriores, el algoritmo de Clasificadores estadísticos (Naive Bayes) tiene la mayor precisión que es 69 % y será elegido para ser implementado dentro de la aplicación web.

4.2. Pruebas utilizando las gráficas conceptuales

En el capítulo 3 se menciona el uso de los algoritmos de minería de texto dentro de la aplicación web para el análisis asistido de los cuestionarios, en esta implementacón se utilizaron a los clasificadores estadísticos para asignar los niveles de competencia a las respuestas del cuestionario de genética.

La aplicación web apoya a los investigadores con la construcción de las gráficas conceptuales (ver sección 3.2.5). Se utilizarán estas gráficas para comparar los resultados del análisis de toda la muestra de los cuestionarios contra una partición en donde se asignará de forma automática el nivel de competencia.

4.2.1. Análisis de gráficas conceptuales

De acuerdo con los investigadores del proyecto, a partir de las gráficas conceptuales es posible observar las representaciones externas que contienen los niveles de competencia en cada muestra.

A partir de estas gráficas comprobaremos nuestra hipótesis: *El modelo propuesto por la metodología del proceso de evaluación asistida es capaz de realizar una evaluación similar a los investigadores*, comparamos las gráficas conceptuales de la evaluación de los investigadores y las gráficas del modelo entrenado. Si las dos muestras generan gráficas conceptuales similares para los mismos niveles de competencia, podemos determinar que ambas asignaciones son equivalentes.

La muestra usada para la asignación del nivel de competencia de forma automática es un cuestionario con 300 respuestas calificadas por los investigadores y 89 respuestas sin nivel de competencia. Se usó el 90 % de las respuestas con nivel de competencia para entrenar al modelo y un 10 % para evaluar al modelo, obteniendo un modelo entrenado con un 79 % de precisión, en la figura 4.12 se muestra el resultado del entrenamiento dentro de la aplicación web.

En la figura 4.13 se muestran dos gráficas conceptuales para el nivel de competencia tres, la gráfica de arriba corresponde a la muestra evaluada por los investigadores y la gráfica de abajo es la muestra evaluada por el modelo, se eligió este nivel, ya que cuenta con el mayor número de respuestas. Es posible observar que la mayoría de

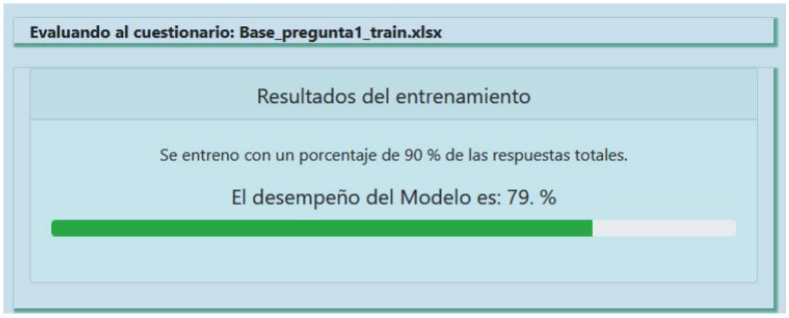


Figura 4.12: Resultados obtenidos al entrenar al modelo con un 90 % de las respuestas evaluadas.

las representaciones externas están presentes en ambas gráficas lo que significa que contiene conceptos similares. Los trigramas presentes en ambos son: genes dominantes recesivos, genes dominantes genes, padre otra mitad, padre mitad madre, madre mitad padre, mitad información génetica, aportan información genética, información genética bebe, información genética padre.

También se muestra en la figura 4.14 las gráficas conceptuales generadas para el nivel de competencia cuatro, en esta comparación se conservan la mayoría de las representaciones externas en ambas muestras, las cuales son: genes dominantes recesivos, mitad información genética, gameto 23 cromosomas, madre otra mitad, padre 23 madre, expresan genes dominantes, 23 padre 23, cromosomas padre 23, 23 cromosomas padre.

Estos resultados muestran que ambas asignaciones contienen representaciones externas similares, lo que indica que el modelo logra generalizar la evaluación de los expertos, lo que puede ayudar a evaluar un cuestionario en un tiempo menor.

Resumen

En este capítulo se analizó el desempeño de los diferentes algoritmos de minería de textos presentados en el capítulo 2, los cuales son: los árboles de decisión, clasificadores estadísticos (Naive Bayes), redes neuronales y máquinas de soporte vectorial, en cada uno de los algoritmos se realizaron varias pruebas con varias particiones de la muestra, para observar la precisión de cada uno de ellos al ser usados con el cuestionario de genética. Considerando que los clasificadores estadísticos lograron el mejor porcentaje de asignación, este algoritmo es implementado en la aplicación web para asignar un nivel de competencia a las respuestas del cuestionario.

Para comprobar nuestra hipótesis: *El modelo propuesto por la metodología del proceso de evaluación asistida es capaz de realizar una evaluación similar a los investigadores*, se comparan las gráficas conceptuales de la evaluación de los investigadores y las gráficas del modelo entrenado. Obteniendo como resultado que la aplicación web es una herramienta que puede apoyar a los investigadores para evaluar de forma parcial un cuestionario de respuestas abiertas, también ayudando a visualizar el contenido de las respuestas en las gráficas y tablas que son creadas automáticamente. Permitiéndole al investigador realizar un análisis del cuestionario de forma sencilla y rápida.

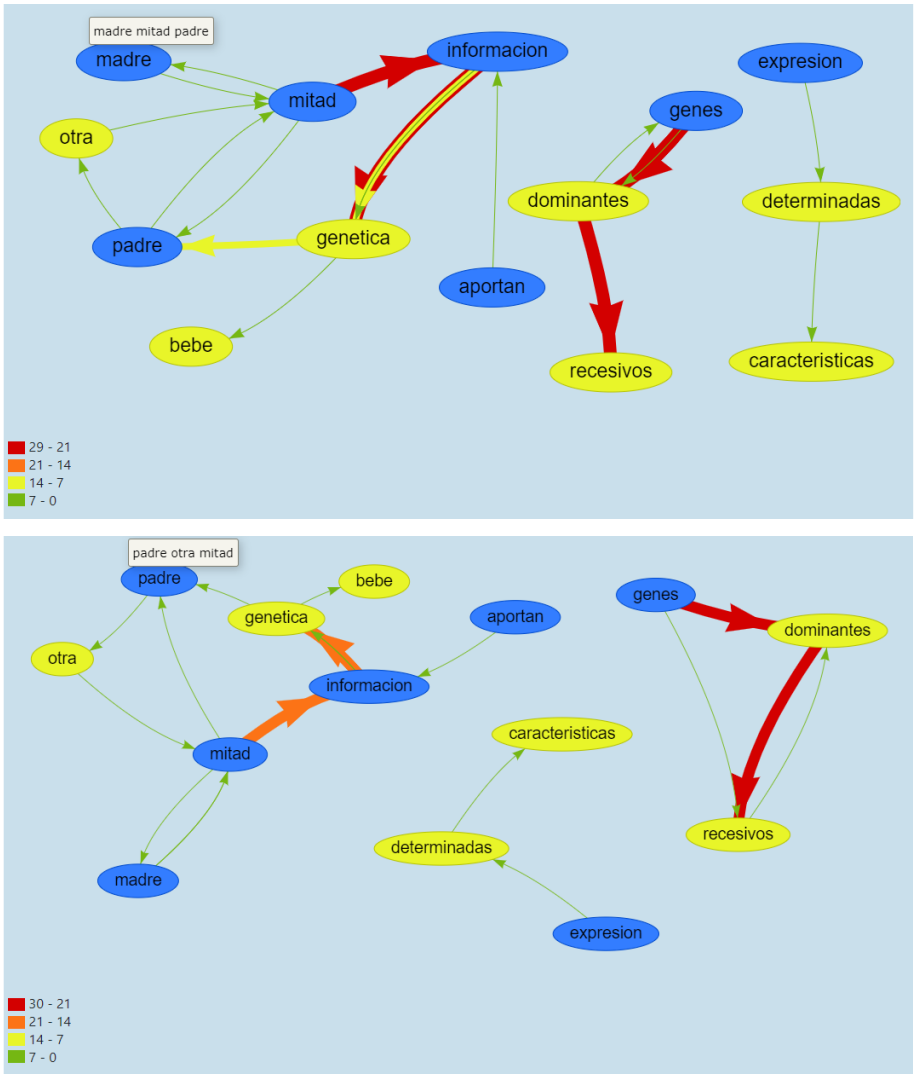


Figura 4.13: Muestra la comparación de las gráficas conceptuales con nivel de competencia tres.

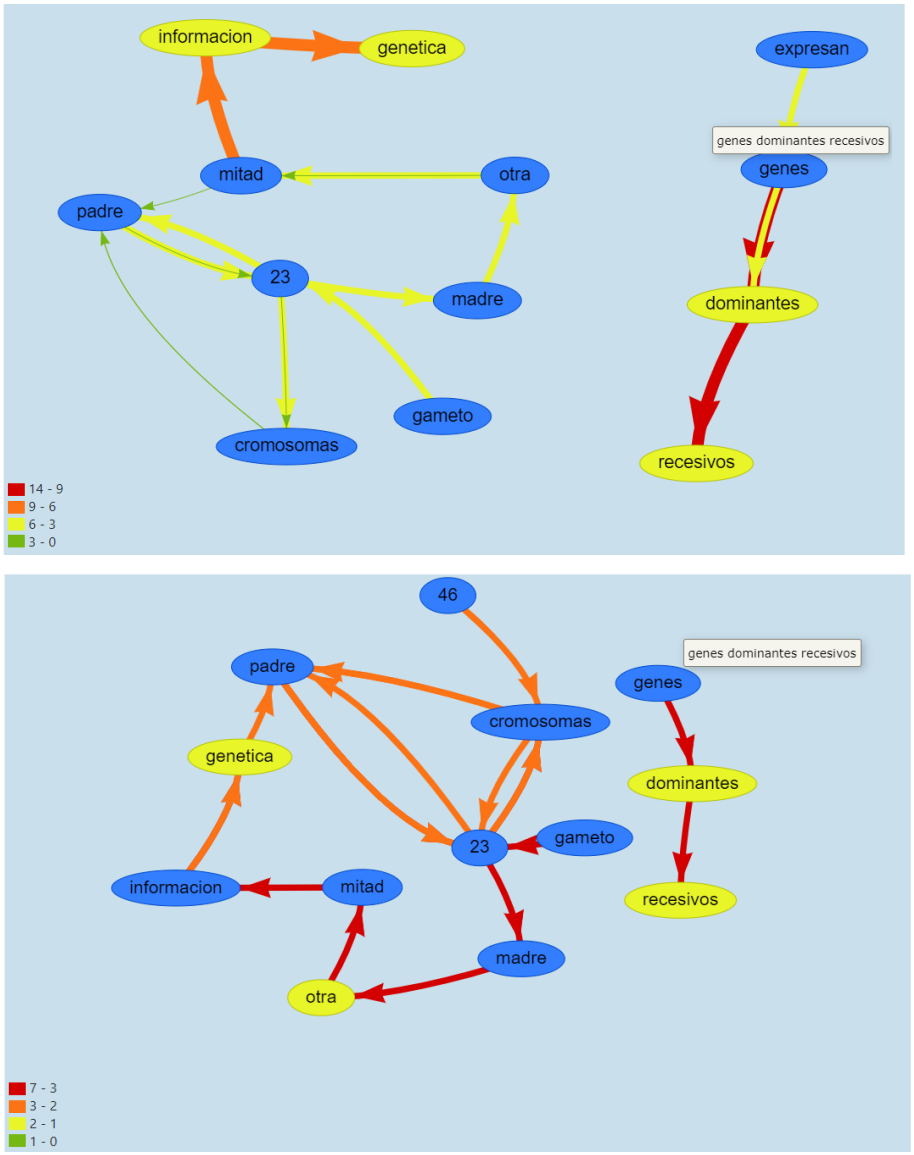


Figura 4.14: Muestra la comparación de las gráficas conceptuales con nivel de competencia cuatro.

Capítulo 5

Resumen general

En esta tesis se presentó una propuesta que usa a las técnicas de minería de texto para apoyar a los investigadores en el proceso de asignación del nivel de competencia en un menor tiempo, dentro del proyecto de investigación "Procesos de transformación de las representaciones científicas en los estudiantes del bachillerato bajo un entorno multi representacional apoyado con tecnologías digitales", centrado en la investigación de la enseñanza de las ciencias en estudiantes de bachillerato.

Los principales temas desarrollados fueron los siguientes:

1. La minería de textos: se comenzó hablando de la minería de datos y la transformación que tienen los datos en cada una de sus etapas. Después se amplió esta visión usando datos textuales y se mencionaron las etapas de la minería de textos, las cuales son: tareas de preprocesamiento, en donde se eliminan caracteres y palabras irrelevantes para el análisis; la aplicación de los algoritmos de minería de texto para buscar los patrones y clasificar a los nuevos documentos; la visualización de la información que permite observar los patrones obtenidos de una manera sencilla y rápida; el refinamiento consiste en aplicar de nuevo alguna tarea previa pero con un cambio en la configuración para obtener el mejor resultado posible.

2. Funcionamiento de algunos los algoritmos de minería de textos: se explicó cómo funcionan los árboles de decisión, los clasificadores probabilísticos, las redes neuronales y las máquinas de soporte vectorial. Este análisis permitió tener un panorama general de los algoritmos y entender sus principales características.

3. Análisis de cuestionarios de preguntas abiertas: se mencionó la transformación de la enseñanza de las ciencias hacia los nuevos enfoques donde se plantea analizar los procesos de construcción y aprendizaje de los conocimientos científicos en los alumnos. Para esto se analizan las representaciones externas que construyen los estudiantes sobre un tema en específico, tales representaciones contienen elementos cognitivos que reflejan los procesos y concepciones de los estudiantes.

   Para observar las representaciones externas de los estudiantes se realizó un cuestionario, en el cual los estudiantes proporcionan una explicación sobre el tema de genética. El cuestionario es evaluado asignando un nivel de competencia para cada una de las respuestas, que señala el nivel de dominio que tiene el alumno sobre el tema.

4. Proceso de evaluación asistida: En la metodología actual, la asignación del nivel de competencia de las repuestas del cuestionario es una tarea manual y puede tomar bastante tiempo. Con el objetivo de apoyar esta tarea se propuso el proceso de evaluación asistida, la cual emplea las técnicas de minería de textos a través de una aplicación web con el fin de otorgar una evaluación automática a respuestas sin un nivel de competencia asignado.

La aplicación web creada como parte de este trabajo, cuenta con funciones básicas para las actividades de los investigadores, como la carga de archivos para el entrenamiento y la creación de un modelo para la evaluación automática. La aplicación permite entrenar varias veces un modelo para encontrar un modelo que se ajuste a la problemática. También cuenta con la opción de visualizar el contenido de las respuestas empleando tablas de frecuencias y gráficas conceptuales, se logró observar que estas gráficas conceptuales están asociadas a las representaciones externas más usadas en las respuestas (mostrando los trigramas más frecuentes y sus relaciones).

La aplicación web crea tablas de frecuencia y gráficas conceptuales por cada pregunta y por cada nivel de competencia del cuestionario analizado. Esto permite observar, en general, las respuestas de una pregunta y observar el comportamiento en cada nivel de competencia.

Con el fin de validar la propuesta del proceso de evaluación asistida se comparó una muestra evaluada por los investigadores con una muestra evaluada por el modelo usando las gráficas conceptuales de la aplicación web.

5.1. Conclusiones

El proceso de evaluación asistida propuesto, tiene el objetivo de apoyar a los investigadores en la tarea de asignación del nivel de competencia, ya que esta tarea es manual y puede tomar bastante tiempo. A partir de este proceso, se construyó un modelo capaz de evaluar o asignar un nivel de competencia automáticamente a las respuestas de los estudiantes. El modelo fue entrenado con una muestra de las respuestas de los alumnos, y este obtuvo un desempeño del 79 % en la asignación de los niveles de competencia. Hay que recordar que la muestra fue evaluada previamente por los investigadores usando los niveles de competencia. Uno de los retos de este trabajo fue aplicar las técnicas de minería de texto dentro del proyecto de investigación para apoyar el análisis de la información. Al realizar esta actividad fueron importantes las habilidades que adquirí en la licenciatura, principalmente en la materia de ingeniería de software, donde su objetivo es dar solución a un problema en específico usando una aplicación, como es el caso de este trabajo.

Para este trabajo se desarrolló la aplicación web, usando el lenguaje de programación Python, ya que este cuenta con bibliotecas para el procesamiento de texto y los algoritmos de aprendizaje automático, las cuales agilizaron la resolución de los problemas. Por otro lado, el framework usado para crear la aplicación web fue Flask, ya que permite crear una aplicación web usando Python. Cabe señalar que hay más frameworks que pueden ser más sencillos para construir una aplicación web, como Django o Web2py, pero dado que cuento con mayor experiencia en Flask que con los otros frameworks, elegí usar este último, para concentrarme en el desarrollo de la propuesta del proceso de evaluación asistida.

Durante el desarrollo de esta propuesta trabajé con dos grupos de investigación del Instituto de Ciencias Aplicadas y Desarrollo Tecnológico (ICAT), los cuales fueron

el ESIE (Espacios y Sistemas Interactivos para la Educación) y el GCDC (Grupo de Cognición y Didáctica de las Ciencias), lo cual fue una gran experiencia, ya que me permitió aprender cómo se realiza el proceso de investigación de la enseñanza de las ciencias y conocer la metodología de diseño centrada en el usuario. Y esto me permitió conocer metodologías más allá de las revisadas en la licenciatura de Ciencias de la Computación. De igual forma entendí que para estas investigaciones es muy importante observar el contenido de las respuestas y facilitar la visualización, por lo cual se han desarrollado estrategias que pueden apoyar a la visualización de la información, como lo son las gráficas conceptuales en la aplicación web.

Estas gráficas conceptuales son construidas por la aplicación web a partir del cuestionario cargado, lo que implica que la aplicación web reconoce el contenido del cuestionario para construir las gráficas conceptuales necesarias.

Así mismo, esta colaboración me mostró, que es necesario realizar un proceso de investigación de la forma de enseñanza, no solo para integrar el uso de nuevos métodos o nuevas tecnologías, sino para tener un entendimiento de las técnicas de enseñanza y cómo se está reflejando este aprendizaje. Además de analizar la posibilidad de obtener mejores resultados agregando el uso de estas nuevas tecnologías.

También como parte del trabajo de esta colaboración con los dos grupos del ICAT, se elaboró un artículo presentando la aplicación web desarrollada y los resultados obtenidos, el cual tiene el nombre de *Web system for text analysis of questionnaire data in science teaching* [9] ,y tuve la oportunidad de presentarlo en el extrajero en el International Conference of Education, Research and Innovation (ICERI) que es un congreso enfocado en la enseñanza, en la misma sesión se presentaron otros proyectos que usaban el aprendizaje automático en la educación, me parecieron interesantes los diferentes enfoques que le dan a estas técnicas, algunas de estas presentaciones estaban enfocadas en la retroalimentación de la enseñanza usando el reconocimiento facial de los estudiantes, otra usaba el reconocimiento de voz para interactuar con ellos, además me parece que este trabajo también da un aporte diferente, ya que refleja el conocimiento adquirido a través del texto.

Considero que lo más interesante de la minería de datos, es su capacidad para el análisis de datos de forma automática, ya que actualmente existe una gran cantidad de información en el mundo, y hay casos en los cuales no se tiene el tiempo necesario para realizar un análisis de forma tradicional. Para este trabajo tuve que profundizar más en los algoritmos de minería, ya que en la licenciatura aprendí las etapas generales como: preprocesamiento de los datos, el uso de los algoritmos y la visualización de la información, pero no conocía a detalle las características de los algoritmos de minería y esto fue importante para la selección del algoritmo que se implementó en la aplicación.

También tuve que explorar otras herramientas de minería de texto, para entender cuáles eran las limitantes del análisis del lenguaje natural y el procesamiento de texto, ya que tenía el reto de usar una cantidad limitada de datos, porque en la licenciatura se suele enseñar la minería de datos con muestras de entrenamiento con más de mil registros o con un tamaño de varios gigabytes de información. Pero en este caso se contaba con una muestra de 387 registros, la cual fue utilizada para entrenar al modelo en la metodología antes mencionada.

Una característica adicional de la propuesta, es que es posible acceder a la aplicación web desde cualquier navegador y no es necesario tener ningún software instalado previamente, solo hay que contar con el cuestionario en formato de Excel. En donde la lista de equivalencias y la lista de stopwords se pueden ajustar de acuerdo con el tema, para que los investigadores puedan personalizar el procesamiento de acuerdo con sus necesidades.

Cabe señalar que la propuesta de evaluación asistida fue diseñada considerando solo el cuestionario del tema de genética, pero esto no limita su uso a este tema, ya que es posible usar cualquier cuestionario que cumpla el formato sin importar su temática. Esto se probó usando un cuestionario de física del mismo grupo de investigadores.

De esta forma, la implementación se aplica a cualquier disciplina donde se necesite analizar el texto de preguntas abiertas.

5.2. Trabajo futuro

A partir de los resultados de esta propuesta se están realizando nuevos trabajos de investigación siguiendo la misma metodología, en estos nuevos trabajos se plantea hacer usos de los algoritmos de agrupación o mejor conocidos como clusters, para realizar un análisis en las tendencias de los cuestionarios, con el fin de observar si existe algún agrupamiento que refleje una perspectiva que no haya sido considerada y que pueda ser de utilidad.

En el uso de otros estudios hay que considerar el tamaño de la muestra y definir los porcentajes que se requieren para entrenar al modelo, recomendando que se puede probar con la mitad de la muestra con su asignación. También considere personalizar las listas de equivalencias y de stopwords para que el entrenamiento sea enfocado en el caso de estudio.

Con respecto a la implementación se recomienda revisar la arquitectura y diseño de la aplicación web, para ofrecer nuevas funcionalidades a los usuario de esta aplicación, ya que esta aplicación web fue construida a la medida del proceso propuesto, ya que en esta etapa del proyecto se buscaba comprobar la utilidad de las tecnicas de mineria de textos al aplicarlas al área de enseñanza de las ciencias y no se tomaron en cuenta otros aspectos como la experiencia del usuario y el uso de metodologías ágiles.

Bibliografía

[1] Fernando Flores Camacho y Beatriz Eugenia García Rivera y Araceli Báez Islas y Leticia Gallegos Cázares. Diseño y validación de un instrumento para analizar las representaciones externas de estudiantes de bachillerato sobre genética. *Revista Iberoamericana de Evaluación Educativa*, 2017.

[2] Jiawei Han, Micheline, and Kamber Jian Pei. *Data mining concepts and techniques*. Elsevier, 2012.

[3] Ricardo Eíto Brun and Jose A. Senso. Minería textual. *El profesional de la información,*, 2004.

[4] Ronen Feldman and James Sanger. *The text Mining Handbook advances approaches in analyzing unstructured data*. Cambridge University prees, 2007.

[5] Raquel Gómez Díaz. *Estudio de la incidencia del conociineto lingüístico en los sistemas de recuperación de la información para el español*. Universidad de Salamanca, España, 2001.

[6] L. Breiman y J. Friedman y R. Olshen y C. Stone. *Classification and Regression Trees*. Chapman and Hall, 1984.

[7] R. y Russell. *Students becoming chemists: developing representational competence*. Amsterdam: Springer, 2005.

[8] Eibe Frank, Mark A. Hall, , and Ian H. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques* . Morgan Kaufmann, Fourth Edition, 2016.

[9] G. De la Cruz Martínez y R. Robles Rios y A. Báez Islas y B.E. García Rivera. *Web system for text analysis of questionnaire data in science teaching*. IATED, 2019.