UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO DE UN SISTEMA DE CATALOGACIÓN DE PRODUCTOS DE E-COMMERCE UTILIZANDO PROCESAMIENTO DE LENGUAJE NATURAL (NLP) Y MACHINE LEARNING

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

RODRIGO ANDRÉS GUERRA LÓPEZ

PROFESOR GUÍA:
MARCELO OLIVARES ACUÑA

MIEMBROS DE LA COMISIÓN:
RICHARD WEBER HAAS
GABRIEL WEINTRAUB YADLIN

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL,
GRADO DE MAGISTER EN GESTIÓN DE OPERACIONES
POR: RODRIGO ANDRÉS GUERRA LÓPEZ
FECHA: ENERO 2019
PROF. GUÍA: MARCELO OLIVARES ACUÑA

DISEÑO DE UN SISTEMA DE CATALOGACIÓN DE PRODUCTOS DE E-COMMERCE UTILIZANDO PROCESAMIENTO DE LENGUAJE NATURAL (NLP) Y MACHINE LEARNING

En el presente proyecto de tesis se aborda el problema de extraer atributos y valores de atributos de productos a partir de descripciones no estructuradas de estos. Por ejemplo, el producto con descripción `'Notebook Lenovo i5 8 GB 500 GB'` posee, entre otros, un atributo `Marca` con valor `'Lenovo'` y un atributo `Memoria RAM` con valor `'8 gigabytes'`. Así, se puede representar un producto p, en un formato estructurado, mediante un conjunto de atributos $\{a_1, a_2, ..., a_n\}$ y un conjunto valores de atributos $\{v_1, v_2, ..., v_n\}$, de la forma $p = \{a_1 : v_1, a_2 : v_2, ..., a_n : v_n\}$.

Para cualquier sitio e-commerce, contar con datos estructurados de los atributos y valores de atributos de sus productos es una ventaja clave. Permite, por un lado, mejorar la experiencia del usuario a través de, por ejemplo, la implementación de filtros de búsqueda por atributos. Por otro lado, permite facilitar y sofisticar cualquier análisis que el sitio quiera realizar a partir de sus datos: desde identificar productos sustitutos hasta segmentar clientes en base a sus preferencias. A pesar de las ventajas que presentan estos datos estructurados, son pocos los e-commerce que cuentan con ellos para todos sus productos. Esto se debe principalmente a que la labor de extraer atributos y valores de atributos es difícil de automatizar, por lo que generalmente se realiza de forma manual. Para un e-commerce con un amplio catalogo de productos, esto se vuelve infactible.

La solución propuesta consiste en un software de aplicación, al cual se denomina *sistema de catalogación*, que hace posible que equipos pequeños de persona puedan mantener grandes catálogos de productos con información estructurada de sus atributos y valores. Para lograr lo anterior, el sistema combina algoritmos de machine learning con técnicas de procesamiento de lenguaje natural (NLP), junto a supervisión humana. A grandes rasgos, el sistema funciona realizando predicciones sobre los atributos y valores de atributos de un producto, las cuales tienen un nivel de confianza asociado. Las predicciones con alto nivel de confianza son validadas automáticamente, mientras que aquellas con bajo nivel de confianza deben ser validadas de forma manual por una persona, mediante preguntas de selección múltiple generadas automáticamente por el sistema. Estas preguntas se diseñan de manera que sean fáciles de responder por cualquier persona con un leve conocimiento del rubro de los productos, lo que permite externalizar la validación manual a plataformas de crowdsourcing.

A través de pruebas realizadas con datos de múltiples e-commerce nacionales, junto a aplicaciones concretas llevadas a cabo en la tienda (e-commerce) de ChileCompra Express, se exhibe el correcto funcionamiento sistema. Gracias a la arquitectura utilizada, que incluye supervisión humana constante, el sistema logra detectar los atributos y valores de atributos de los productos con una alta precisión de sobre un 80 %.

Tabla de Contenido

Introducción

Vocabulario

Es necesario definir ciertos conceptos que serán utilizados a lo largo del presente trabajo[1]. Se define un *producto* como cualquier artículo que pueda ser comercializado por un retailer. Un *atributo* es una característica que describe cierta propiedad de un producto. Algunos ejemplos de atributos son: `Marca`, `Color` y `Peso`. Una *categoría* es un conjunto de productos que poseen los mismos atributos. Son ejemplos de categoría `Notebooks`, `Pendrives`, `Televisores`. Un *valor de atributo* es un valor especifico que toma un determinado atributo. Por ejemplo, `'Verde'`, `'Rojo'` y `'Azul'` son posibles valores del atributo `Color`. Una *tabla de atributos* corresponde a una tabla de dos columnas, en donde la primera contiene los atributos de un producto y la segunda los valores que estos toman en el producto (Tabla 1). Se procede a formalizar la definición anterior:

Definición. Tabla de atributos: Sea p un producto cualquiera, con atributos $\{a_1, a_2, ..., a_n\}$ y valores de atributos $\{v_1, v_2, ..., v_n\}$, se dirá que la tabla de atributos del producto p corresponde a cualquier estructura de datos capaz de representar a p como:

$$p = \{a_1 : v_1, a_2 : v_2, \ldots, a_n : v_n\}$$

ATRIBUTO	VALOR DE ATRIBUTO
Marca	Lenovo
Procesador	Intel Core i5
Tamaño de Pantalla	15.6 pulgadas
Memoria RAM	8 gigabytes
Almacenamiento	1 terabyte
Sistema Operativo	Windows 10 Pro

Tabla 1: Ejemplo de tabla de atributo para un producto de la categoría `Notebooks`.

Tablas de atributos en e-commerce

Para cualquier e-commerce, contar con una tabla de atributos para cada uno de los productos de su catálogo genera múltiples beneficios. Permite mejorar tanto la experiencia de

[1]Las definiciones de producto, atributo y valor de atributo son extraídas de [15]

los clientes al navegar por el portal, como el entendimiento del e-commerce sobre sus productos, sus clientes y el comportamiento de estos últimos. Algunos ejemplos concretos de estos beneficios son:

- **Permiten la implementación de filtros de búsqueda.** Para implementar un buen sistema de filtros de búsquedas, es necesario contar con tablas de atributos para los productos. Estos filtros facilitan enormemente el proceso de búsqueda de productos para los potenciales clientes, ya que les permiten reducir rápidamente el conjunto de productos mostrados al conjunto de productos de su interés. Esta mejora a la experiencia de compra se ve reflejada fuertemente en las ventas: el año 2013, la consultora estadounidense Econsultancy publicó un articulo en el que afirman que la implementación de filtros en el e-commerce buyakilt (actualmente Kilt Society[2]) aumentó en un 26 % la tasa de conversión y en un 76 % los ingresos de la tienda [16].
- **Permiten calcular similitud entre productos.** Calcular métricas de similitud entre productos es una tarea compleja cuando se utilizan fuentes de datos no estructuradas como texto plano. Al utilizar texto plano, dos productos que son exactamente el mismo pueden ser identificados como completamente distintos solo por la forma en la que están escritos. Por ejemplo, los productos `'Lenovo 8 gigabytes 1000 gigabytes 15,6 pulgadas Windows 10 Pro'` y `'Lenovo(r) 15.6'' 8GB 1TB W10Pro'` presentan los mismos atributos (Figura 1), pero la similitud de texto entre ambos es baja. Si se cuenta con tablas de atributos para los productos, se puede definir una métrica de similitud de forma sencilla, como por ejemplo, el porcentaje de atributos en los que ambos productos tienen el mismo valor (100 % para el caso del ejemplo). Una métrica de similitud como la anterior identificar fácilmente grupos de productos similares e idénticos, esto es importante para, por ejemplo, el diseño de algoritmos de recomendaciones de productos.
- **Permiten realizar análisis robustos de los datos.** Al hacer cualquier análisis empírico de datos de un e-commerce, por ejemplo transaccionales, es vital poder definir a los productos como una serie de atributos (o tablas de atributos). En general, contar con tablas de atributos para los productos simplificará en gran medida cualquier análisis de minería de datos que se quiera realizar (utilizando estos datos).

Lenovo 8 gigabytes 1000 gigabytes 15,6 pulgadas Windows 10 Pro

Marca:	Lenovo
Disco Duro:	1 terabyte
Memoria RAM:	8 gigabytes
Tamaño Pantalla:	15.6 pulgadas
Sistema Operativa:	Windows 10 Pro

Producto 1

Lenovo(r) 15.6" 8GB 1TB W10Pro

Marca:	Lenovo
Disco Duro:	1 terabyte
Memoria RAM:	8 gigabytes
Tamaño Pantalla:	15.6 pulgadas
Sistema Operativa:	Windows 10 Pro

Producto 2

Figura 1: Par de productos distintos a nivel de texto, pero con los mismos valores de atributos en una representación estructurada.

A pesar del valor que entregan estos datos, es común que los e-commerce no presenten información de sus productos en formato de tablas de atributos, o bien, la presenten pero de manera incompleta [8, 10]. Esto último quiere decir que la tabla no presenta todos los

[2] `https://kiltsociety.com`

atributos y valores de atributos necesarios para caracterizar al producto. Este fenómeno, por lo general, no se debe a que no se cuente con la información de los atributos de los producto, ya que esta información suele ser presentada de forma implícita tanto en formato de imágenes como texto. Tampoco se debe a que los e-commerce no estén interesados contar con tablas de atributos para sus productos, ya que como se menciona anteriormente, son múltiples los beneficios que estas entregan.

La ausencia de tablas de atributos en e-commerce se debe principalmente a que son costosas de obtener. Por lo general, las tablas de atributos se construyen procesando y estructurando información de un producto desde fuentes de datos no estructurados como imágenes y/o texto. A este proceso se le denominará *catalogar* (Figura 2). Como se explicará más adelante, catalogar es un proceso difícil de automatizar, por lo que generalmente se realiza de forma manual [8]. Esto quiere decir que una o más personas analizan imágenes y/o texto con información de un producto, extraen información de los atributos y valores de atributo del producto y posteriormente estructuran esta información en un formato de tabla de atributos. Para un e-commerce con un catálogo de productos reducido, catalogar manualmente resulta factible, pero a medida que el tamaño del catálogo aumenta, en cantidad de productos , la labor se hace cada vez más difícil de realizar, hasta llegar al punto de volverse infactible: estudios anteriores muestran que una persona puede catalogar aproximadamente 13 productos por hora [20]. Si se considera una jornada laboral de 8 horas, entonces una persona puede catalogar alrededor de 100 productos diarios. Tomando el caso del Walmart, que tiene un catálogo de decenas de millones de productos, con una tasa de crecimiento de aproximadamente 100 mil productos diarios [20], se necesitaría un equipo de alrededor de 1000 personas cuya única función fuese catalogar para poder mantener los productos con sus respectivas tablas de atributos actualizadas. Claramente, contratar a un número tan elevado de personas para realizar esta labor no resulta rentable.

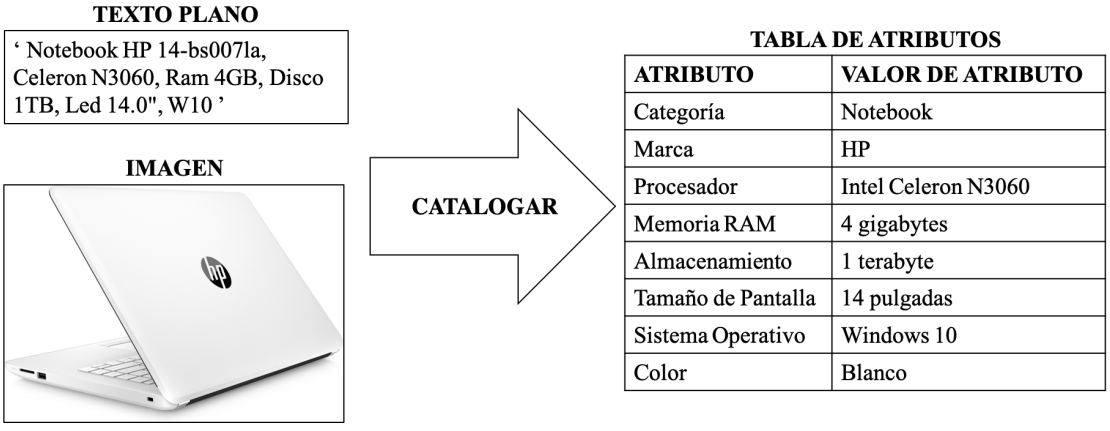

Figura 2: Catalogar es el proceso de extraer y estructurar información de los atributos y los valores de atributos de un producto a partir de fuentes de datos no estructurados como imágenes y/o texto.

ChileCompra

El presente trabajo de tesis nace dentro de un proyecto FONDEF IDeA con la Dirección de Compra y Contratación Pública (ChileCompra), organismo estatal que busca generar ahorro, eficiencia y transparencia en el mercado público nacional. Actualmente en el país, los organismos públicos realizan una parte significativa de sus compras a través de la tienda online ChileCompra Express[3]. ChileCompra Express consiste en un portal web que funciona como un e-commerce marketplace, en donde proveedores del estado ofrecen sus productos y servicios, previa adjudicación de un proceso licitatorio, e instituciones públicas –o que administren fondos públicos– realizan sus compras. Los montos transados por la tienda ChileCompra Express superan los 2250 millones de dolares por año [12].

Por lo general los productos de la tienda ChileCompra Express no cuentan con una tabla de atributos completa. Esto quiere decir que las tablas de atributos no tienen definido una cantidad de atributos que permita identificar completamente a los productos. Por ejemplo, en el rubro de alimentos, las leches poseen solo 3 atributos: Marca (e.g. Colún), Medida (e.g. 1 litro) y Envase (e.g. caja). Estos son insuficientes para identificar una leche, ya que se desconocen otros atributos relevantes, como el sabor (e.g. chocolate), si contiene lactosa o no, o si es entera, semidescremada o descremada. ChileCompra maneja un catálogo de más de 100 mil productos, los cuales son catalogados de forma manual [12]. Como se mencionó anteriormente, para un volumen de productos elevado, catalogar de forma manual resulta infactible.

Para ChileCompra, contar con información estructurada de sus productos es tan o incluso más relevante que para un e-commerce común. Un estudio anterior realizado a ChileCompra [12] concluye que mejorar la catalogación de los productos[4] permitirá reducir los precios transados en la plataforma ChileCompra Express, generando así ahorro en el gasto público. Son dos las principales formas en las que una mejor catalogación permite reducir los precios transados:

- **Permite guiar a los compradores públicos a proveedores más baratos.** Existe evidencia que el diseño de una plataforma e-commerce influye significativamente en la decisión de compra de sus clientes. Experimentos realizados en el marketplace eBay [6] han demostrado mediante A/B testing que cambios en la plataforma orientados a priorizar las ofertas más baratas en los resultados de las búsqueda, permiten reducir los precios transados en hasta 16 %. Se espera que cambios similares en la plataforma de ChileCompra Express puedan traducirse en reducciones de precios transados. Sin embargo, estos cambios requieren contar con una métrica de similitud entre productos capaz de identificar productos idénticos y similares con exactitud. Como se explico anteriormente, las tablas de atributos permiten definir métricas confiables de similitud entre productos.
- **Permiten sofisticar el diseño de licitaciones.** Es evidente que el diseño de una licitación afectará la adjudicación. Si el diseño de la licitación falla en promover la competencia, entonces los productos se adjudicarán a un mayor precio. Esto implicará

[3] `http://www.mercadopublico.cl`

[4] 'Mejorar la catalogación de los productos' es equivalente a 'Generar mejores tablas de atributos para los productos'

que el catálogo de productos de la tienda ChileCompra Express estará formado de productos más caros, lo que finalmente se traduce en mayores precios transados. En el diseño de una licitación es vital conocer los productos a nivel de atributo para poder definir correctamente los grupos de productos, idénticos o similares, que competirán en el proceso licitatorio. Si los productos no están definidos mediante tablas de atributos, entonces la tarea de identificar estos grupos de productos se dificulta considerablemente. Debido a lo anterior, es muy probable que productos (o grupos de productos) que debiesen competir entre si queden en grupos distintos y, por lo tanto, no compitan. Esto último hace que la competencia disminuya [12].

Automatizar el proceso de catalogar

Se busca diseñar y entrenar un método/algoritmo/sistema capaz de automatizar –de manera parcial o total– la extracción de tablas de atributos en base a descripciones no estructuradas de productos. A estas descripciones, se les llamará *nombre de producto*. Un ejemplo de nombre de producto es `'Lenovo V330 15.6'' Core i5 8GB 1TB W10 Pro'`. Por motivos de brevedad, de ahora en adelante, cada vez que se mencione el proceso de *catalogar* se estará haciendo referencia al proceso de "catalogar a partir de un nombre de producto".

Catalogar automáticamente no es una tarea sencilla de automatizar. Se presentan a continuación algunas de las dificultades que conlleva esta esta tarea:

- **Requiere extraer tanto los atributos como valores de atributos**: La solución propuesta debe ser capaz de detectar no solo los valores de atributo de un producto, sino que también sus atributos, lo que dificulta considerablemente el problema. A modo de ejemplo se considera el producto con nombre `'Lenovo V330 15.6'' Core i5 8GB 1TB W10 Pro'`. Identificar a partir del texto que `'Core i5'` y `'W10 Pro'` corresponden a valores de atributos no es una tarea demasiado compleja. Esto se conoce como un problema de *keyphrase extraction* para lo cual existen diversos métodos, como los presentados en [9]. Por otro lado, inferir además los atributos a los que corresponden estos valores, en este caso `Procesador` y `Sistema Operativo` respectivamente, añade un grado de dificultad extra al problema.
- **Los valores de atributos pueden escribirse de múltiples formas**: Considerando el caso del atributo `Almacenamiento`, presente en un producto de la categoría `Notebooks`. Un posible valor de este atributo es `'1 terabyte'`. Este valor suele ser escrito de distintas formas: `'1 TB'`, `'1000 GB '`, `'1000 gigabytes'`, etc. La solución propuesta debe ser capaz de detectar las múltiples formas de escribir un valor de atributo y homologarlas a un solo valor. Por ejemplo, en el caso anterior, si definimos `'1 terabyte'` como valor de atributo, entonces la solución deberá detectar que el producto con nombre `'Lenovo V330 15.6'' Core i5 8GB 1TB W10 Pro'` posee un atributo `Almacenamiento` con valor igual a `'1 terabyte'` (y no `'1TB'`).

Solución propuesta

La solución propuesta consiste en un software de aplicación que permite automatizar gran parte de la catalogación de productos. A esta aplicación se le denominará *sistema de catalogación* o simplemente *sistema*. El sistema de catalogación es del tipo semiautomático, lo que quiere decir que "efectúa parte de su funcionamiento de manera automática tras una ayuda manual" [1].

El sistema cuenta con un *proceso de catalogación*, el cual, a partir del nombre de un producto y mediante el uso de algoritmos de procesamiento de lenguaje natural y de aprendizaje automático (machine learning), es capaz de formular predicciones de pares *{atributo: valor}* para un producto. Estas predicciones tendrán asociadas un nivel de confianza, normalizado entre 0 y 1. Las predicciones con un nivel de confianza sobre cierto umbral serán validadas de forma automática, mientras que aquellas predicciones cuya confianza no superen dicho umbral, serán transformadas a un formato de pregunta de selección múltiple, como por ejemplo:

El producto con nombre: `'Lenovo V330 15.6'' Core i5 8GB 1TB W10 Pro'`, ¿tiene un atributo `Memoria RAM` con valor igual a `'8 gigabytes'`?

Estas preguntas, deberán ser respondidas de forma manual por personas. La información de la respuesta, en este casos `Si` o `No`, será incorporada a la base de datos del sistema. De está forma el sistema se mantendrá constantemente aprendiendo de sus aciertos y errores (supervisión constante).

Sin ser el objetivo original, el sistema funciona como una herramienta de administración de catálogos, en la que se pueden crear, modificar o eliminar categorías, atributos, productos y valores de atributos. La innovación del sistema es que permite catalogar, i.e. determinar los atributos y valores de atributos de un producto, rapidamente haciendo uso del proceso de catalogación, que automatiza gran parte de esta labor.

En el primer capítulo de este trabajo se describe el diseño del sistema, incluyendo el modelo de datos utilizado y describiendo el proceso de catalogación. En los capítulos 2 y 3 se profundiza en los subprocesos que realizan y validan o rechazan las predicciones en las cuales se basa el sistema. En el capítulo 4 se presenta el prototipo desarrollado del sistema, junto a un manual de uso. Finalmente en el capítulo 5, se presentan dos aplicaciones para las cuales la utilización del sistema de catalogación puede beneficiar a ChileCompra.

Capítulo 1

Descripción del sistema

El *sistema de catalogación* es diseñado con el objetivo principal de automatizar –de manera parcial o total– el proceso de extracción de *tablas de atributos* a partir de descripciones no estructuradas de productos, a las que se denominarán *nombres de productos* (Figura 1.1).

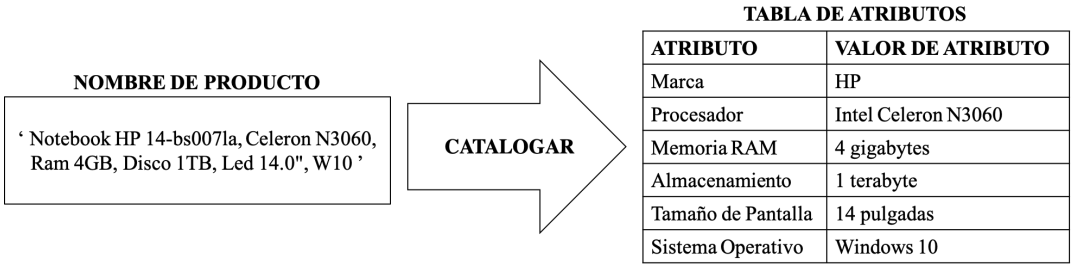

Figura 1.1: *Catalogar* es el proceso de extraer una *tabla de atributos* a partir de un *nombre de producto*.

En la primera sección del capítulo se presenta el *modelo de datos* bajo el cual se construye la base de datos que almacena toda la información del sistema, entre ella, la información de los productos y sus tablas de atributos. Para poder catalogar automáticamente en el sistema, es necesario ingresar cierta información a la base de datos de forma manual. A este proceso se le denomina *inicialización del sistema* y se describe en la segunda sección del capítulo. En la tercera y última sección del capítulo, se presenta el *proceso de catalogación*, que corresponde al proceso que permite catalogar productos de forma semiautomática.

1.1. Modelo de Datos

El modelo de datos es diseñado para cumplir dos objetivos: *(i)* ser capaz de almacenar la información de las tablas de atributos de los productos y *(ii)* ser capaz de estructurar esta información en un formato que permita alimentar de datos a los algoritmos de predicción del *proceso de catalogación*. Es por este último objetivo que ciertas decisiones tomadas para el diseño del modelo de datos pueden parecer algo arbitrarias de momento. Estas decisiones

serán abordados cuando se presente el *proceso de catalogación*. Se comenzará presentando una versión simplificada del modelo de datos para facilitar la comprensión de la versión completa del modelo, la cual será presentada a continuación de esta última.

1.1.1. Versión simplificada

Se construye un modelo de datos relacional basado en 4 clases: **Producto**, **Atributo**, **Valor de Atributo** y **Categoría**. Estas clases modelan los conceptos definidos al comienzo de este informe:

> "Se define un *producto* como cualquier artículo que pueda ser comercializado por un retailer. Un *atributo* es una característica que describe cierta propiedad de un producto. Algunos ejemplos de atributos son: `Marca`, `Color` y `Peso`. Una *categoría* es un conjunto de productos que poseen los mismos atributos. Son ejemplos de categoría `Notebooks`, `Pendrives`, `Televisores`. Un *valor de atributo* es un valor especifico que toma un determinado atributo. Por ejemplo, `'Verde'`, `'Rojo'` y `'Azul'` son posibles valores del atributo `Color`."

Las 4 clases base y sus relaciones son presentadas en la Figura 1.2 y explicadas a continuación:

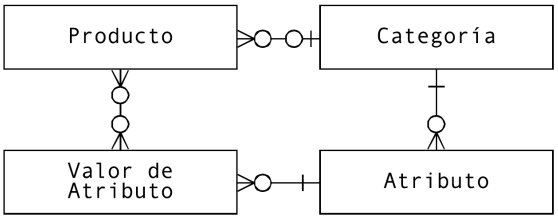

Figura 1.2: Diagrama de clases para las 4 clases bases.

- **Categoría - Atributo:** las categorías de productos, poseen cero o más atributos. Un ejemplo es la categoría `Notebooks`, que posee atributos como `Marca`, `Procesador`, `Tamaño de Pantalla`, etc. Un atributo está relacionado a solo a 1 categoría. Si es que existe otra categoría con atributo `Marca`, por ejemplo `Pendrives`, este atributo corresponderá a un objeto nuevo en la clase **Atributo**, distinto al atributo `Marca` de la categoría `Notebooks`.
- **Producto - Categoría:** un producto puede pertenecer a lo más a una categoría (puede también no pertenecer a ninguna). Por ejemplo, el producto con nombre `'Notebook Asus 15.6'' Core i5 8GB 1TB'` deberá pertenecer a la categoría `Notebooks`. Una categoría puede contener 0 o más productos. Cuando un producto pertenece a una categoría pasa a heredar sus atributos.
- **Atributo - Valor de Atributo:** un atributo puede tomar varios valores de atributos. Por ejemplo, el atributo `Marca` de la categoría `Notebooks` puede tomar valores como `'Lenovo'`, `'Toshiba'`, etc. De la misma forma, el atributo `Tamaño de Pantalla` (también de la categoría `Notebooks`) puede tomar valores como `'13 pulgadas'`, `'15 pulgadas'`, etc. Un valor de atributo está asignado a solo un atributo. Esto quiere decir que si, por ejemplo, existe otro atributo con valor de atributo `'Lenovo'`, como

podría ser el atributo `Marca` de la categoría `Tablets`, entonces este corresponderá a otra entrada en la clase **Valor de Atributo**.

- **Producto - Valor de Atributo:** los productos tendrán asignados 0 o más valores de atributos. Recordar que un valor de atributo está asociado a exactamente 1 atributo. Luego, si un producto de la categoría `Notebooks` tiene asociado el valor de atributo `'Lenovo'`, se puede recuperar a que atributo corresponde este valor de atributo (`Marca`). Un valor de atributo puede estar asignado a 0 o más productos.

Se puede notar que el modelo de datos simplificado es recursivo. La relación entre **Producto** y **Categoría** podría ser eliminada y la integridad del sistema se mantendría. Sin embargo, como se verá más adelante, el *proceso de catalogación* funciona prediciendo la categoría a la que pertenece un producto, para así descubrir sus atributos y, finalmente, utilizando la información de los atributos, poder predecir los valores que estos toman en el producto. Por la razón anterior, se hace necesario mantener la relación entre las clases **Producto** y **Categoría** y construir un modelo recursivo.

1.1.2. Versión completa

La versión completa del sistema está basada en la versión simplificada. Las diferencia están principalmente en que la clase **Atributo** se divide en dos clases, **Atributo Nominal** y **Atributo Ratio**. La clase **Valor de Atributo** también es dividida en dos clases, **Valor de Atributo Nominal** y **Valor de Atributo Ratio**. Esta división se realiza porque ambos tipos de atributos (y valores de atributos) tienen distintas propiedades. Al separar ambos tipos se puede diseñar una estructura de datos que se adecúe a las particularidades de cada uno estos. En la versión completa del modelo de datos también se añaden tablas que permiten estructurar los datos de cada uno de los dos tipos de atributos y valores de atributos.

Se procede a definir los conceptos de *atributos nominales* y *atributos ratio*. Los atributos nominales son aquellos que toman valores discretos que no pueden ser ordenados en una escala inherente, mientras que los atributos ratio toman valores compuestos por una magnitud numérica y una unidad de medida. Los atributos ratio si pueden ser ordenados en una escala. Así, en la categoría `Notebooks`, el atributo `Memoria RAM` es ratio, tomando valores `4 gigabytes`, `8 gigabytes`, etc. mientras que el atributo `Marca` es nominal, con valores discretos como `'Lenovo'`, `'HP'`, etc.[1]

Atributos y valores de atributos nominales

El modelo de datos debe considerar el hecho de que un valor de atributo nominal puede ser representado de múltiples formas. Por ejemplo, `'HP'` y `'Hewlett Packard'` y son dos formas distintas de representar un mismo valor de atributo nominal. Esto se modela añadiendo una nueva clase: **Token**. Se puede pensar en un token como "una forma distinta de escribir un valor de atributo". Un token puede ser una palabra o un grupo de 2 o más palabras. Al añadir

[1] La nomenclatura de atributo nominal y atributo ratio se basa en la presentada en [19].

la clase `Token`, se puede modelar el caso anterior definiendo a `'HP'` como valor de atributo nominal y a `'Hewlett Packard'` como token relacionado con el valor de atributo `'HP'`.

A continuación, se presenta el diagrama de clases entre `Atributo Nominal`, `Valor de Atributo Nominal` y `Token` (Figura 1.3), y posteriormente se describen las relaciones entre estas.

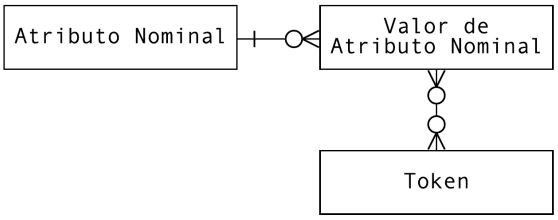

Figura 1.3: Diagrama de las clases de `Atributo Nominal`, `Valor de Atributo Nominal` y `Token`.

- **Atributo Nominal - Valor de Atributo Nominal:** Estas clases se relacionan de la misma forma que las clases `Atributo` y `Valor de Atributo` en el la versión simplificada del modelo de datos.
- **Valor de Atributo Nominal - Token:** Un token corresponde a un fragmentos de texto que actúa como sinónimo de un valor de atributo nominal. Un valor de atributo nominal está relacionado a uno o más tokens y un token está relacionado a uno o más *valores de atributos nominal*.

Atributos y valores de atributos ratio

Al igual que los atributos nominales, un atributo ratio se puede representar de múltiples formas. Consideremos, por ejemplo, el valor de atributo ratio `'1 terabyte'`. Este valor de atributo también puede ser escrito: *(i)* utilizando una abreviatura, o otra forma distinta de representar la unidad, por ejemplo `'1 TB'`; o *(ii)* utilizando otra unidad, por ejemplo `'1000 gigabytes'`. Para considerar lo anterior en el modelo de datos, se añaden dos clases `Unidad` y `Dimensionalidad`.

Una *unidad* corresponde a una unidad de medida, como por ejemplo, `gigabyte` o `pulgada`. Una *dimensionalidad* corresponde a un conjunto de unidades de una misma medida. Por ejemplo, las unidades `gigabytes`, `megabyte` y `terabyte` pertenecen a la dimensionalidad `tamaño de datos`; mientras que las unidades `pulgada`, `metro` y `centímetro` pertenecen a la dimensionalidad `longitud`. Para cada dimensionalidad existe una unidad base. Por ejemplo, se puede definir a `metro` como la unidad base de `longitud`. Todas las unidades –como elementos de la clase **Unidad**– cuentan con un campo numérico, denominado *multiplicador*, que indica por cuanto se debe multiplicar la unidad para transformarla a unidad base. Por ejemplo, si se define `metro` como unidad base, el multiplicador de la unidad `kilometro` es `1000`. Solo puede existir una unidad base por dimensionalidad. Una unidad también cuenta con una lista de abreviaturas o sinónimos. Por ejemplo, la unidad `mililitro` también se escribe como: `ml`, `centimetro cúbico` o `cc`.

El diagrama de clases de entre **Atributo Ratio**, **Valor de Atributo Ratio**, **Unidad** y **Dimensionalidad** es presentados en la Figura 1.4. Las relaciones entre las clases son descritas a continuación

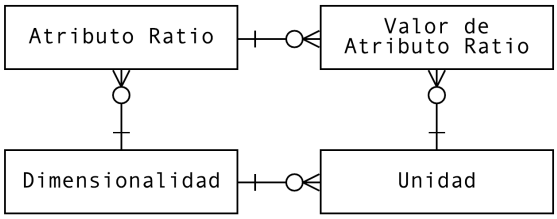

Figura 1.4: Diagrama de clases de Atributo Ratio y Valor de Atributo Ratio.

- **Atributo Ratio - Valor de Atributo Ratio:** Se relacionan de la misma forma que las clases `Atributo` y `Valor de Atributo` de la versión simplificada del modelo de datos.
- **Dimensionalidad - Unidad:** Una unidad pertenece solo a una dimensionalidad y una dimensionalidad puede contener cero o más unidades.
- **Atributo Ratio - Dimensionalidad:** Todo *atributo ratio* estará asociado a una y solo una *dimensionalidad*. De esta forma, en la categoría `Notebooks`, los atributos ratio `Disco Duro` y `Memoria RAM` estarán asociados a la dimensionalidad `tamaño de datos`, así como el atributo `Tamaño de Pantalla` estará relacionado a la dimensionalidad `longitud`.
- **Valor de Atributo Ratio - Unidad:** Los *valores de atributo ratio*, por su parte, estarán relacionados a una y solo una *unidad*. Esta *unidad* deberá pertenecer a la misma *dimensionalidad* que el *atributo ratio* al cual está asociado dicho *valor de atributo*. A modo de ejemplo, el *atributo ratio* `Disco Duro` de la categoría `Notebooks` puede tomar *valores de atributo ratio* como `‘500 [gigabytes]’`[2], `‘1 [terabyte]’`, `etc.`, pero no `‘5 [gramos]’` o `‘10 [litros]’`.

Versión completa

El diagrama de clases del modelo de datos completo es presentado en la Figura 1.5. En el Apéndice A se presenta una versión mas detallada del diagrama de clases, incluyendo los atributos (de las clases) y las clases intermedias.

1.2. Inicialización del sistema

Antes de poder ejecutar el *proceso de catalogación* es necesario añadir cierta información a la base de datos del sistema. En concreto se deben definir las categorías que existirán en el catálogo y los respectivos atributos de cada una. Esto requiere conocimiento del rubro de los

[2]Los *valores de atributo ratio* serán representados como una magnitud numérica seguida por su unidad (objeto de la clase **Unidad**) entre corchetes.

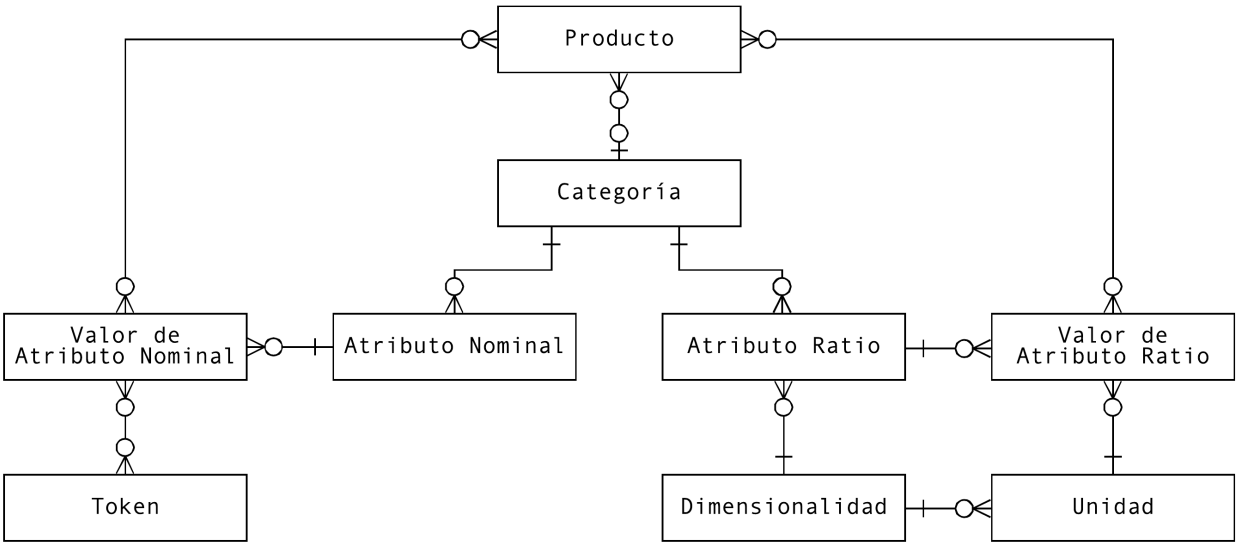

Figura 1.5: Diagrama de clases completo.

productos que se comercializaran. No se considera posible extraer esta información desde los propios datos de los productos, por lo que se sigue un enfoque manual. Una persona deberá ingresar las categorías y sus atributos a la base de datos, mediante la interfaz gráfica del sistema. A modo de ejemplo para el desarrollo de este informe, se supondrá que se definen las categorías: `Notebooks`, `Pendrives` y `Televisores`, con atributos como se muestra en la Figura 1.6. Tanto las categorías como los atributos pueden ser creados, modificados o eliminados en cualquier momento a través de la interfaz gráfica del sistema.

Notebooks

(nominales)

- Marca (‘Lenovo’)
- Procesador (‘Intel Core i5’)
- Sistema Operativo (‘Windows 10 Pro’)

(ratios)

- Memoria RAM (‘8 gigabytes’)
- Almacenamiento (‘1 terabyte’)
- Tamaño de Pantalla (‘15 pulgadas)’

Pendrives

(nominales)

- Marca (‘SanDisk’)
- Interfaz (‘USB 3.0’)

(ratios)

- Capacidad (’32 gigabytes)

Televisores

(nominales)

- Marca (‘Samsung’)
- Tipo HD (‘ Ultra HD’)
- Smart TV (‘Si’)

(ratios)

- Tamaño de Pantalla (‘72 pulgadas’)

Figura 1.6: Ejemplos de categorías definidas en el sistema y sus atributos. Se indica si los atributos son del tipo nominal o ratio. Se presenta entre paréntesis un ejemplo de valor para cada atributo.

Si bien el enfoque seguido requiere definir manualmente las categorías y los atributos de estas, esta tarea solo consume una fracción pequeña del tiempo que consume catalogar productos manualmente. Para un gran e-commerce, la relación entre el número de categorías y el número de productos es de alrededor de $1 : 20$ [20], es decir, por cada categoría existen aproximadamente 20 productos. Por lo anterior, al definir las categorías y sus atributos de forma manual se mantiene factible el proposito del sistema, siempre que este, sea capaz de automatizar la catalogación de productos, que es lo que más tiempo consume de realizar

manualmente.

Definir las categorías y los atributos que estarán presentes en el sistema, además de ser una tarea difícil de automatizar, es bastante subjetiva. Por ejemplo, uno podría argumentar que los atributos de la categoría `Televisores` deberían incluir también el atributo `Peso` o incluso, que la categoría debería dividirse en dos: `Televisores No Smart TV` y `Televisores Smart TV`. Aqui no existe una respuesta correcta, pero considerando el diseño del sistema y los algoritmos del *proceso de catalogación* se entregan tres recomendaciones:

- **Definir las categorías necesarias para cubrir el universo de productos del catálogo:** un producto pertenecerá a una de las categorías definidas o no pertenecerá a ninguna. Un producto que no pertenece a ninguna categoría no podrá ser catalogado y, por lo tanto, no se tendrá información ni de sus atributos ni de sus valores de atributos.
- **Definir categorías a nivel de tipo de producto:** según el modelo de datos, todos los productos de una categoría tendrán los mismos atributos, por lo que no hace sentido definir categorías como conjuntos de productos con distintos atributos. En otras palabras, no se deben crear categorías del tipo `Notebooks e Impresoras`, ya que corresponden a una agrupación de distintos tipos de productos, con distintos atributos. Lo correcto sería definir dos categorías: `Notebooks`, con atributos `Memoria RAM`, `Tamaño de Pantalla`, etc., e `Impresoras` con atributos como `Tecnología de Impresion`, `Blanco y Negro o Color`, etc.
- **En cada categoría, definir los atributos que se consideren necesarios para identificar correctamente a un producto:** Si se definen pocos atributos, se pierde información que puede ser relevante para la identificación de un producto. Por ejemplo, si no se define el atributo `Tamaño de Pantalla` en la categoría `Notebooks`, no se almacenará información sobre este atributo. Se considera que el atributo `Tamaño de Pantalla` si es relevante para caracterizar un `Notebook` y, por lo tanto, debería ser agregado. Por otro lado, definir un número excesivo de atributos puede ser inútil si es que estos no aparecen en las descripciones de los productos. Por ejemplo, si bien `Consumo Energético` podría ser perfectamente considerado como un atributo de `Notebooks`, no aparece en los nombres de producto y, por lo tanto, aunque se defina, no será un atributo detectado en la mayoria de los productos.

1.3. Proceso de catalogación

El sistema cuenta con un *proceso de catalogación* (Figura 1.7), que combina algoritmos de machine learning con técnicas de procesamiento de lenguaje natural para automatizar en gran parte la detección de los atributos y los valores de atributo de un producto a partir de su nombre de producto. Cuando el proceso de catalogación concluye, este retorna una tabla de atributos para el producto, es decir, un listado de los atributos del producto con sus repectivos valores (Figura 1.1).

El proceso de catalogación funciona de forma secuencial. En una primera etapa –*Subproceso Detectar Categoría*– se encarga de detectar la categoría del producto en cuestión. Una vez conocida la categoría del producto, también se conocerán sus atributos del producto, ya que

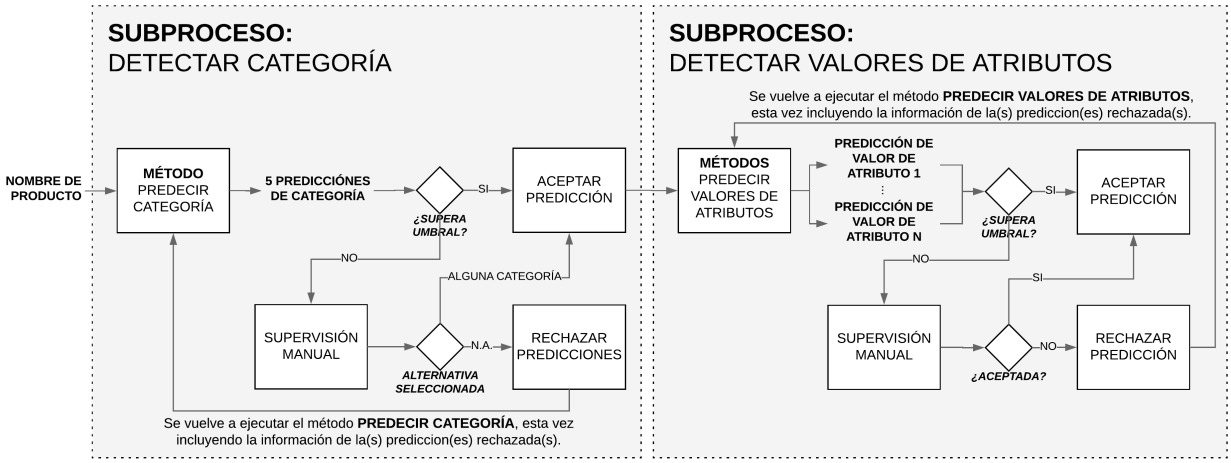

Figura 1.7: Proceso de catalogación.

estos heredan la categoría. Al conocer los atributos del producto se puede ejecutar la segunda etapa, llamada *Subproceso Detectar Valores de Atributos*, que se encarga de determinar los valores que toman los atributos del producto.

El Subproceso Detectar Categoría comienza con el *Método Predecir Categoría*. Este método procesa el nombre del producto y retorna una lista de 5 predicciones de categoría. Para lograr lo anterior, se utiliza un método de clasificación (kNN) entrenado en los productos previamente categorizados (i.e. productos para los cuales se conoce su categoría). Las predicciones de categoría retornadas por el método cuentan con nivel de confianza asociado entre 0 y 1, donde 1 es el máximo nivel de confianza. Si la predicción de categoría con mayor nivel de confianza tiene un nivel de confianza mayor a cierto umbral (e.g. 0.75), entonces se aceptará la predicción de forma automática y se ingresará a la base de datos del sistema que el producto pertenece a dicha categoría. En el caso contrario –que la predicción con mayor nivel de confianza no supere el umbral– el producto no sera asignado a ninguna categoría momentáneamente y se generará una pregunta de selección múltiple a partir de las 5 predicciones de categoría entregadas por el Método Predecir Categoría (Figura 1.8). Esta pregunta deberá ser respondida manualmente una persona. Una vez que la pregunta sea respondida, se conocerá la categoría del producto, a no ser que la respuesta seleccionada sea `'Ninguna de las anteriores'`. En dicho caso, se volverá a ejecutar el Método Predecir cCtegoría, esta vez incluyendo la restricción de que el producto no puede pertenecer a ninguna de las 5 categorías anteriormente rechazadas. El Subproceso Detectar Categoría finaliza una vez que se descubre la categoría del producto. Este subproceso es detallado en el Capítulo 2.

Una vez que se conoce la categoría del producto y, por lo tanto, también sus atributos, se ejecuta el *Subproceso Detectar Valores de Atributos* que comienza con el *Método Predecir Valores de Atributos*. Este método predice los valores para los atributos del producto a partir de su nombre de producto utilizando algoritmos de procesamiento de texto entrenados en los datos almacenados en la base de datos del sistema. El método retorna entre 0 y n predicciones de valores de atributo por cada atributo del producto ($n > 1$). Una predicción de valor de atributo –al igual que una predicción de categoría– tiene asociado un nivel de confianza. Las predicciones que superen cierto umbral, que puede ser distinto al de las predicciones

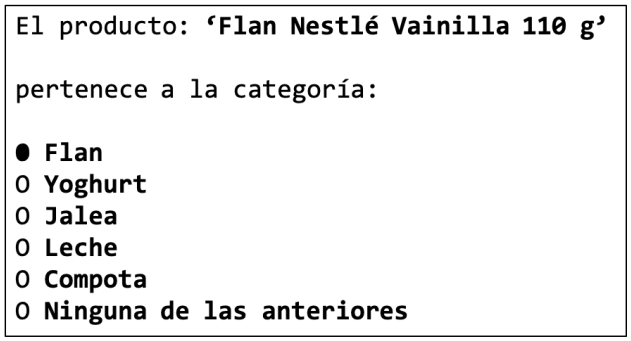

Figura 1.8: Ejemplo de validación de predicciones de categoría.

de categoría, serán agregadas al sistema de forma automática. Para las predicciones que no superen el nivel de confianza, se formularán preguntas de validación que deberán ser respondidas manualmente por personas, al igual que las validaciones de categorías (Figura 1.9). En el Capítulo 3 se profundiza en el Subproceso Detectar Valores de Atributos.

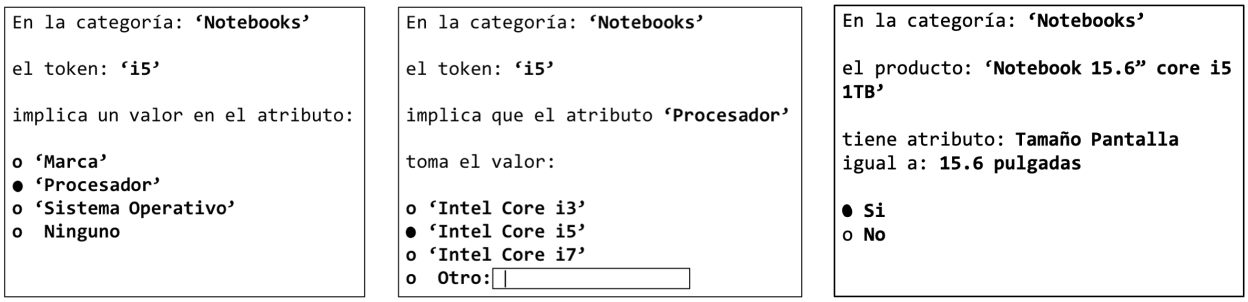

Figura 1.9: Ejemplos de validación de predicciones de atributos.

En la Figura 1.10 se presenta un ejemplo del funcionamiento del proceso de catalogación sobre un método nombre de producto: el Subproceso Detectar Categoría valida automáticamente que el producto pertenece a la categoría `Notebooks`, ya que la predicción tiene un nivel de confianza 0,89, lo que (supongamos) está sobre el valor del umbral. Como se conoce la categoría del producto, se conocen también sus atributos, ya que heredan de la categoría. Utilizando los atributos, se ejecuta el método detectar valores de atributos, el cual retorna una serie de predicciones de valores para los distintos atributos del producto. Dependiendo del umbral, algunas de ellas pueden ser validadas automáticamente y otras de forma manual.

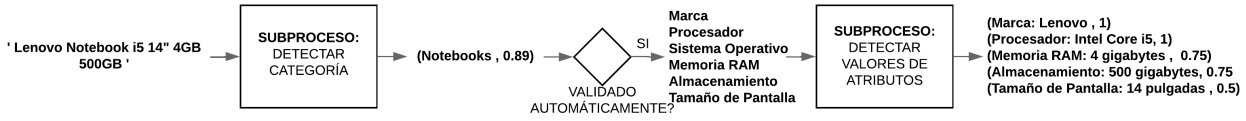

Figura 1.10: Ejemplo del *proceso de catalogación* en un producto.

Capítulo 2

Subproceso Detectar Categoría

En el siguiente capítulo se hace foco en el primer subproceso del proceso de catalogación (Figura 2.1), que tiene como objetivo detectar la categoría a la que pertenece un producto y, de esta forma, también sus atributos.

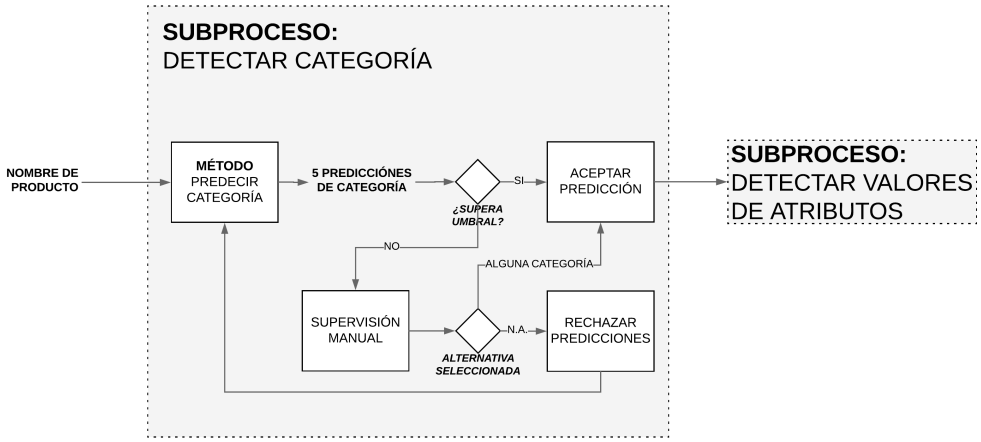

Figura 2.1: Proceso de catalogación, Subproceso Detectar Categoría

El subproceso funciona resumidamente como sigue: inicia con el *Método Predecir Categoría*, el cual procesa un nombre de producto y retorna una lista de entre 1 y 5 predicciones de categoría, cada una con un nivel de confianza asociado[1] (Figura 2.2). Si de esta lista, la predicción con mayor nivel de confianza supera un cierto umbral, se acepta la predicción y el subproceso termina. En el caso contrario, una persona deberá validar la categoría correcta del producto, seleccionándola del listado de 5 predicciones entregado por el Método Predecir Categoría. En el caso que ninguna de estas sea la categoría correcta, la persona puede rechazar rechazarlas todas. Según las pruebas realizadas, la categoría correcta se encuentra en este listado en un 94 % de los casos. Si la persona valida una de las 5 predicciones categorías, entonces se acepta la predicción y termina el subproceso. Si la persona rechaza las 5 predicciones, entonces se vuelve a ejecutar el subproceso, esta vez con la información de que el producto no pertenece a ninguna de las categorías previamente rechazadas (i.e. se genera

[1]El nivel de confianza de las predicciones está normalizado entre 0 y 1.

un nuevo listado de predicciones de categoría). En la primera sección del capítulo se profundiza en el funcionamiento del Método Predecir Categoría y en el diseño de las preguntas de validación manual.

Tanto los algoritmos como los parámetros utilizados en el Subproceso Detectar Categoría fueron seleccionados mediante la realización de pruebas, escogiendo aquellos que entregan mejores resultados. El valor Las distintas alternativas probadas, los resultados de las pruebas y los criterios de selección son presentadas en la segunda parte del capítulo.

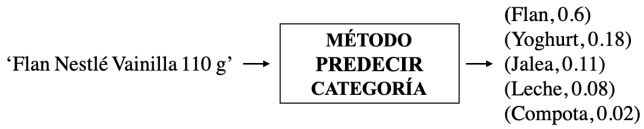

Figura 2.2: Ejemplo de lista de predicciones de categorías entregadas por el Método Predecir Categoría al procesar la descripción no estructurada de un producto.

2.1. Descripción del Subproceso Detectar Categoría

2.1.1. Método Predecir Categoría

El Método Predecir Categoría funciona de la siguiente manera:

Inicialmente, el método se encarga de transformar las descripciones no estructuradas de los productos a un listado de fragmentos de texto preprocesados. Para esto, se procesan los textos de las descripciones transformándolos a minúsculas, eliminando caracteres especiales (e.g. '®', '&') y removiendo acentos gráficos (tildes). Luego, se **tokeniza** la descripción preprocesada utilizando los **unigrams** como unidades de texto (Figura 2.3).

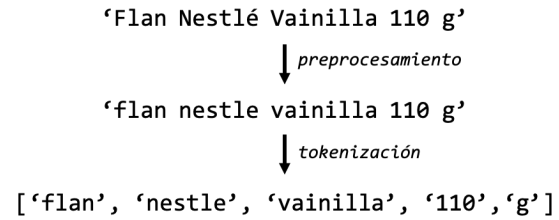

Figura 2.3: Ejemplo de preprocesamiento y tokanización para la descripción de un producto

Posteriormente, se vectoriza el listado de tokens que describen al producto. Esto quiere decir que ahora cada producto estará representado por un vector numérico en un espacio de alta dimensionalidad. Para esto se utiliza el modelo **bag-of-words (BOW)** (Figura 2.4). Los vectores resultantes de este modelo son ponderados por factores calculados mediante **term frequency–inverse document frequency (tf-idf)**, con el propósito de mejorar la representación vectorial. A la representación vectorial de un producto se le denominará *vector producto*

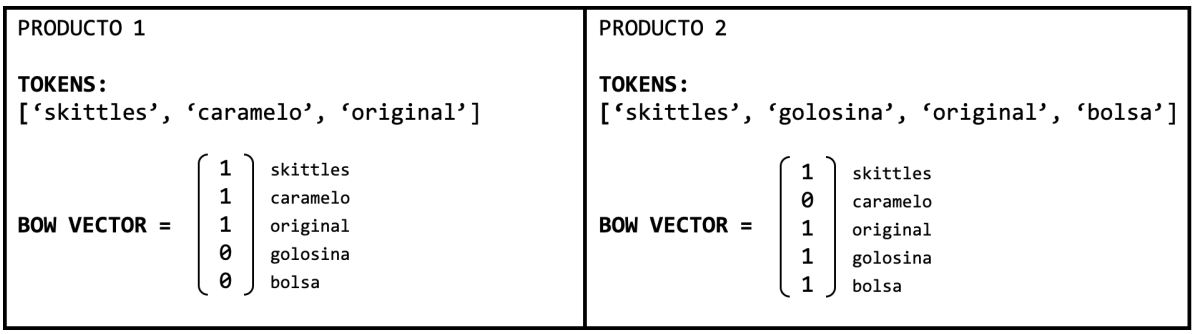

Figura 2.4: Ejemplo de vectorización mediante bag-of-words.

Luego, el método divide los productos y sus respectivos vectores en dos grupos: *(i)* aquellos para los cuales se conoce su categoría y *(ii)* aquellos para los que no. Se procede a predecir las categorías del segundo grupo de productos haciendo uso del algoritmo de clasificación supervisada **k-nearest neighbors (kNN)**, con $k = 5$, entrenado sobre el primer conjunto de productos. El método predecir categoría requiere que al menos 1 producto esté categorizado en cada categoría para poder funcionar.

KNN retorna un listado de posibles categorías, cada una con un nivel de confianza asociado. Existen diversas formas de calcular la confianza de una predicción para el algoritmo kNN, de las cuales se selecciona la confianza de Cover [3] por entregar mejores resultados. De este listado, se seleccionan las primeras 5 con mayor nivel de confianza (el resto se desprecian). Si el listado es menor a 5, se conservan todas las predicciones.

2.1.2. Proceso de Validación

Si del listado de predicciones de categoría entregado por el Método Predecir Categoría, al procesar un nombre de producto, aquella con mayor nivel de confianza supera cierto umbral, entonces se validará automáticamente que dicho producto pertenece a la categoría entregada por esta predicción. El valor del umbral es crítico para el funcionamiento del sistema, por lo que se realizan pruebas para determinar su valor óptimo, concluyendo un valor de 0.6.

En el caso contrario que la predicción de mayor confianza no supere el umbral, se deberá validar la categoría del producto de forma manual a través de una pregunta de selección múltiple generada automáticamente por el sistema. Esta pregunta es del tipo `¿A que categoría pertenece el producto` x`?`, donde `x` corresponde al nombre de producto. Como alternativas se muestran las categorías del listado de predicciones Método Predecir Categoría (que contiene a lo más 5 predicciones) y la alternativa extra `Ninguna de las anteriores` (Figura 2.5).

Si quien responde la pregunta de validación selecciona una categoría como respuesta, entonces se validará que el producto pertenece a dicha categoría. Si selecciona `Ninguna de las anteriores`, entonces se volverá a ejecutar el *Método Categorizar*, esta vez añadiendo la restricción de que el producto no puede pertenecer a ninguna de las categorías anteriormente rechazadas.

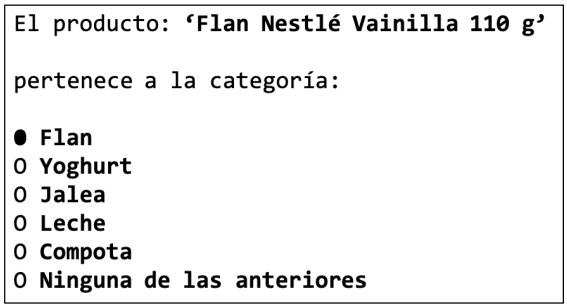

Figura 2.5: Ejemplo de pregunta de validación de categoría para el producto del ejemplo de la Figura 2.2.

2.2. Selección de algoritmos y parámetros

A lo largo de esta sección se argumentarán las decisiones tomadas para seleccionar los algoritmos y parámetros que conforman el Subproceso Detectar Categoría. La mayoría de estas decisiones están basadas en análisis que se desprenden de los resultados obtenidos al realizar pruebas sobre datos de catálogos de productos.

Los *datos de prueba* corresponden a 23,181 productos provenientes de 2 tiendas e-commerce nacionales, obtenidos mediante **web scraping**. Una de estas tiendas comercializa productos del rubro tecnológico (e.g. notebooks, televisores), mientras que la segunda vende productos alimenticios y para el hogar (e.g. lácteos, papel higiénico). Se presenta un análisis más detallado de los datos en el Apéndice B.

Para todas las evaluaciones realizadas en esta sección se utilizaron 4 validaciones cruzadas aleatorias (random permutation cross-validation) con un 60 % de los productos como conjunto de entrenamiento (training set) y un 40 % como conjunto de evaluación (test set).

2.2.1. Método de clasificación (kNN)

Se decide fijar **kNN** como método de clasificación principalmente por dos razones:

(i) **Ser robusto antes desbalances de clases.** En este caso, una clase corresponde a una categoría. Un desbalance de clases ocurre cuando en un catálogo existen categorías con muchos productos y otras con muy pocos. Este fenómeno ocurre de forma frecuente en los catálogos y se ha demostrado que **kNN** tiene un desempeño decente en estos casos [17, 21]. Un análisis a los datos de prueba muestra que lestos también presentan desbalance de clases (Apéndice B).

(ii) **Existen múltiples métricas de confianza asociadas al método.** Como kNN es un método simple y altamente interpretable, se han definido múltiples métricas de confianza para las predicciones a lo largo de la historia. El articulo [4] presenta 8 formas de calcular la confianza de una predicción. Para el correcto funcionamiento del sistema, es muy importante obtener una métrica de confianza que se aproxime lo más posible a la probabilidad real de que la predicción sea correcta, ya que permite separar

correctamente las predicciones que serán validadas automáticamente de aquellas que necesitaran validación manual. Contar con varias formulas para calcular la confianza de una predicción entrega libertad para ponerlas a prueba y seleccionar aquella que entregue mejor desempeño.

2.2.2. Vectorización de descripciones de producto (BOW tf-idf) y parametros kNN (k=5)

El objetivo es lograr representar los productos como vectores en algún espacio de alta dimensionalidad en el que productos de una misma categoría ocupen posiciones similares.

Se ponen a prueba dos metodologías distintas de vectorización de productos: **bag-of-words (BOW)** y **average-word2vec (avg-w2v)**. Para cada una de estas, se utilizan distintos tipos de preprocesamiento y **tokenización** de texto; y distintas combinaciones de parámetros. A continuación se presentan las distintas alternativas.

- **Preprocesamiento y tokenización:** como preprocesamiento se prueba *(i)* transformar el texto a minúsculas, *(ii)* eliminar acentos gráficos, *(iii)* remover caracteres no alfanuméricos[2] y *(iv)* eliminar números. Se **tokeniza** utilizando **unigrams** y se prueba si añadir **bigrams** mejora los resultados.

- **Parametros bag-of-words:** se prueba si ponderar los vectores por factores **tf-idf** mejora los resultados.

- **Parametros avg-w2v:** se prueban distintas modalidades de entrenar **word2vec (w2v)** (skipgram y CBOW) y también distintos números de dimensiones (20, 40, 80, 160, 320). También se analiza si las representaciones mejoran al ponderar los vectores por factores **tf-idf**.

Para determinar la mejor representación vectorial, se procede a clasificar los vectores resultantes de cada representación utilizando **kNN** con distintos valores de k. Se selecciona aquel modelo cuya representación vectorial entregue un mayor **accuracy** en esta clasificación. Si se cumple el objetivo, que es lograr que productos de una misma categoría estén representados por vectores similares, entonces el porcentaje de productos clasificados correctamente deberá ser alto. Por lo tanto, entre mejor sea la representación, mayor debería ser este valor (**accuracy**).

Resultados y Análisis

Para **BOW** los mejores resultados se obtienen al preprocesar las descripciones de los productos transformándolas a minúscula y eliminando solo caracteres especiales. Añadir **bigrams** empeora los resultados, por lo que se utiliza solo **unigrams**.

[2]alfanumérico: formado por letras y números

Por otro lado, para **avg-w2v**, los mejores resultados se obtienen al preprocesar las descripciones de los productos solo transformándolas a minúscula. Se utilizan solo **unigrams** ya que añadir **bigrams** empeora los resultados. En cuanto a los parámetros de **w2v**, se obtienen mejores resultados al fijar el número de dimensiones en 40 y utilizando CBOW como método de entrenamiento.

En la Tabla 2.1 se presentan los resultados obtenidos para las mejores versiones de los modelos de vectorización **BOW** y **avg-w2v**, con y sin aplicar la transformación **tf-idf**.

	BOW	**BOW tf-idf**	**avg-w2v**	**avg-w2v tf-idf**
k = 3	0.692	0.723	0.619	0.618
k = 5	0.699	**0.731**	0.613	0.612
k = 10	0.685	0.722	0.596	0.597
k = 15	0.671	0.713	0.580	0.583
k = 20	0.654	0.700	0.567	0.568

Tabla 2.1: Accuracies obtenidos para distintas representaciones vectoriales y distintos valores de k.

Se desprende que **BOW** con transformación **tf-idf** domina al resto de los modelos de vectorización, para todo k, por lo que es elegido para vectorizar las descripciones de los productos. Para este modelo de vectorización, se obtienen los mejores resultados al fijar el parámetro k de **kNN** igual a 5.

2.2.3. Métrica de confianza (Cover)

Se exploran distintas métricas de confianza para las predicciones del método **kNN**. Para esto, se utiliza la publicación de Dalitz, 2009 [4], que resume varias métricas. En particular, se estudian las confianzas de Cover [3], Droettboom [7] y Dasarathy [5], las cuales son descritas a continuación.

- **Cover:** Sea x una observación a clasificar y ω_{i} una categoría cualquiera i. Sea k el número total de vecinos y k_{i} el número de vecinos que pertenecen a la categoría i. La confianza de Cover queda definida como,

$$P(x \in \omega_{\mathrm{i}}) = k_{\mathrm{i}}/k \tag{2.1}$$

  La confianza de Cover está normalizada entre 0 y 1, tomando el valor 1 cuando todos los vecinos pertenecen a la categoría predicha.

- **Dasarathy:** Sea x la observación a clasificar y ω_{i} una categoría cualquiera. Se define $\mathrm{d}(\cdot)$ como la distancia euclidiana entre dos puntos. Sea y una observación del conjunto de entrenamiento. La confianza de Dasarathy queda definida como,

$$P(x \in \omega_{\mathrm{i}}) = \left(1 - \frac{min_j\{\mathrm{d}(x, y_j) | y_j \in \omega_{\mathrm{i}}\}}{min_j\{\mathrm{d}(x, y_j) | y_j \notin \omega_{\mathrm{i}}\}}\right) \tag{2.2}$$

El numerador de la fracción corresponde a la distancia entre la observación x y su vecino más cercano perteneciente a la categoría ω_i. Por otro lado, el denominador corresponde a la distancia entre la observación x y su observación más cercana que no pertenezca a la categoría w_i. Esta métrica está normalizada entre 0 y 1, y es altamente susceptible a outliers en el conjunto de entrenamiento.

- **Droettboom:** Sea x la observación a clasificar y ω_i una clase cualquiera. Sea y una observación del conjunto de entrenamiento y N el conjunto de todas las observaciones del conjunto de entrenamiento. La confianza de Droettboom queda definida por,

$$P(x \in \omega_i) = \left(1 - \frac{min_j\{d(x, y_j)|y_j \in \omega_i\}}{max_j\{d(x, y_j)|y_j \in N\}}\right) \tag{2.3}$$

El numerador de la fracción se puede interpretar como la distancia entre la observación x y su vecino más cercano de la clase ω_i, mientras que el denominador corresponde a la distancia la observación x y su vecino más lejano (de cualquier clase). La confianza de Droettboom entrega valores entre 0 y 1, y es aplicable incluso cuando $k = 1$ o cuando existen clases atípicas con solo 1 observación clasificada en el set de entrenamiento.

.

Para evaluar las distintas métricas de confianza se construye un vector *hit* para el conjunto de evaluación, que toma el valor de 1 si la categoría predicha por el método es la categoría real del producto y el valor de 0 si no. La intuición señala que la correlación entre la métrica de confianza y *hit* debería ser positiva, ya que a mayor confianza, mayor debiese ser la probabilidad de que la predicción sea correcta.

Luego, el criterio utilizado para seleccionar la mejor métrica de confianza será el **coeficiente de correlación de Pearson** entre la métrica y *hit*: entre más cercana a 1 sea la correlación, mejor será la métrica.

Resultados y Análisis

Los **coeficientes de correlación de Pearson** entre hit y las distintas métricas de confianzas son presentadas en la Tabla 2.2. La correlación entre *hit* y la métrica de Cover es indiscutiblemente mayor al resto, por lo que es seleccionada como métrica de confianza a utilizar.

	Coeficiente de correlación
Cover	0.525
Dasarathy	0.298
Droettboom	0.225

Tabla 2.2: Coeficiente de correlación de Pearson entre las métricas de confianza y *hit*.

2.2.4. Número de predicciones para la validación manual (n=5)

Como se menciona anteriormente, si para un producto su predicción de categoría con mayor nivel de confianza no supera cierto umbral, entonces se deberá validar de forma manual la categoría a la que pertenece este producto. La validación manual se realiza a través de preguntas de selección múltiple (Figura 2.5).

El *Método Categorizar* entrega una lista de predicciones de categoría para un producto, cuyo tamaño máximo es 5 (dado que se fija $k = 5$). Al ordenar esta lista de forma descendiente con respecto al nivel de confianza se puede definir donde "cortar" el listado de predicciones para seleccionar las *top n* con mayor nivel de confianza (Figura 2.6).

```
TOP 1
(Flan, 0.6)
```

```
TOP 3
(Flan, 0.6)
(Yoghurt, 0.18)
(Jalea, 0.11)
```

```
TOP 5
(Flan, 0.6)
(Yoghurt, 0.18)
(Jalea, 0.11)
(Leche, 0.08)
(Compota, 0.02)
```

Figura 2.6: Ejemplo de top 1, top 3 y top 5 predicciones con mayor nivel de confianza para el producto de la Figura 2.2.

La pregunta que se busca responder es de que tamaño debe ser la lista de categorías presentadas como alternativas en la validación manual. Se analizan los casos de presentar las *top 3* y *top 5* categorías con mayor nivel de confianza.

Existe un **tradeoff** al aumentar el número de alternativas. Por un lado, aumenta la probabilidad de que la categoría correcta esté presente entre estas. Pero, por otro lado, al aumentar el número de alternativas también se aumenta la dificultad de la pregunta para quien la responde [20].

Se toma una decisión en base al porcentaje de veces que la categoría correcta aparece en el listado de *top n* predicciones (**accuracy**). No se cuenta con una métrica que mida el esfuerzo cognitivo al que se somete quien responde, por lo que se hace complejo estimar la dificultad de una pregunta al variar el número de alternativas que se presentan.

Resultados y Análisis

En la Tabla 2.3 se presentan los porcentajes de veces que la categoría correcta aparece en el listado de *top n* para distintos valores de n.

accuracy top 1	accuracy top 3	accuracy top 5
0.731	0.905	**0.925**

Tabla 2.3: Accuracies obtenidos al considerar las *top n* predicciones de categorías.

Se decide fijar $n = 5$ debido a que entrega un aumento de 2 % en el **accuracy** comparado

con $n = 3$. Se considera –de forma subjetiva– que este aumento en el accuracy es más relevante que el aumento en la dificultad al incrementar el número de alternativas de 3 a 5.

2.2.5. Umbral validación automática (*0.6*)

Según el valor del umbral se definirá cuales productos serán categorizados de forma automática y cuales serán categorizados mediante una pregunta de validación manual.

Este valor es crítico para el funcionamiento correcto del sistema. Si el umbral es bajo, entonces la mayoría de los productos serán clasificados de forma automática, pero, ¿serán clasificados correctamente?. A medida que se baja el valor del umbral, se aceptan predicciones de categoría con un menor nivel de confianza asociado y, por lo tanto, con una mayor probabilidad de ser incorrectas.

Si al contrario, se fija un valor elevado de umbral, se asegura que los productos clasificados de forma automática tengan un alto nivel de confianza y, por lo tanto, se disminuye la probabilidad de cometer errores. Pero, por otro lado, se incrementa el número de productos que deberán ser clasificados de forma manual, lo cual tiene un costo monetario asociado.

Para determinar un valor óptimo para el umbral se toman en cuenta dos puntos:

- el porcentaje de productos clasificados de forma automática que son clasificados correctamente (*precisión*).
- el porcentaje de productos clasificados de forma automática del total de productos.

Resultados y Análisis

Como era de esperar, existe un **tradeoff** entre la precisión del clasificador y el porcentaje de productos clasificados. Si se fija un umbral alto, e.g. 0.9, se asegura una precisión de 0.99, pero solo se lograría clasificar un 28 % de los productos de forma automática, siendo el resto enviado a supervisión manual (Figura B.2).

Hay que tener en cuenta que el Método Predecir Categoría se retro-alimenta tanto de las validaciones automáticas como de las validaciones manuales, por lo que parece más riesgoso fijar un valor de umbral bajo a uno alto. Si se utiliza un valor de umbral bajo, se cometerá un número mayor de errores de clasificación de productos y, al sistema retro-alimentarse de estos errores, la calidad de las predicciones futuras se verá mermada.

Teniendo en consideración lo anterior y los resultados obtenidos (Figura B.2), se decide fijar el valor del umbral en 0.6, lo que asegurará una precisión de 0.94 y permitirá catalogar automáticamente un 47 % de los productos (enviando 53 % restante a revisión manual).

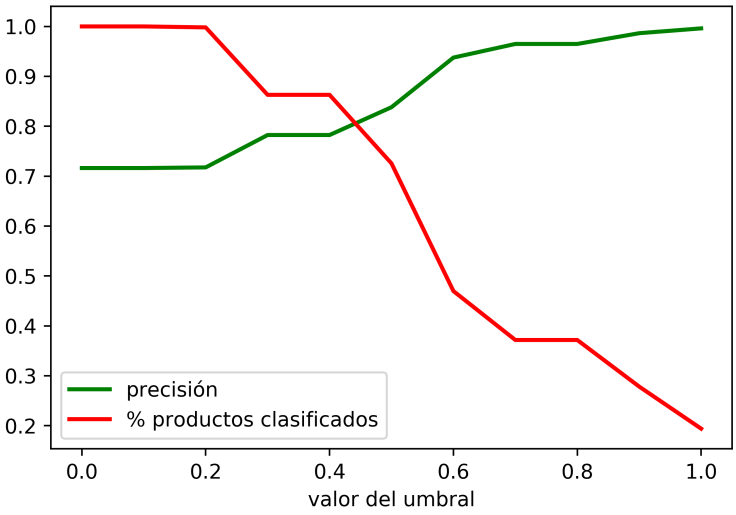

Figura 2.7: **Precisión** y porcentaje de productos clasificados (del total de productos) al variar el valor del umbral. La métrica de confianza utilizada corresponde a la de Cover.

Capítulo 3

Subproceso Detectar Valores de Atributos

En el siguiente capítulo se detalla el subproceso *Detectar Valores de Atributos* (Figura 3.1, encargado de extraer los valores que toman los atributos de un producto a partir de su *nombre de producto*. El subproceso requiere conocer los atributos del producto a priori, por lo que se ejecuta solo cuando el subproceso *Detectar Categoría* haya finalizado. Recordar que al conocer la categoría de un producto, se conocen también sus atributos, ya que estos heredan de la categoría.

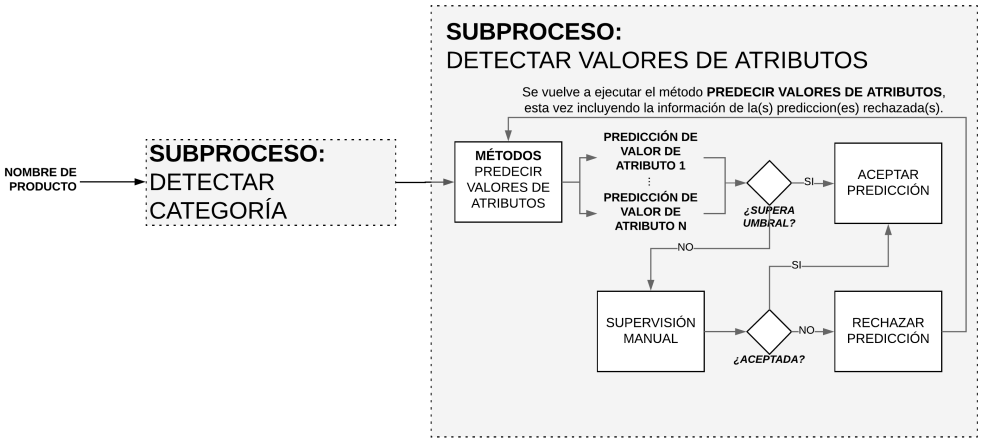

Figura 3.1: Proceso de catalogación, subproceso detectar Valores de atributos.

El diseño del subproceso Detectar Valores de Atributos es similar al del subproceso Detectar Categoría. Este comienza con el método *Predecir Valores de Atributos*, el cual retorna predicciones de valores para los atributos del producto que se está catalogando. En concreto, el método retorna una, varias o ninguna predicción de valor de atributo para cada atributo del producto. Si para un atributo, su predicción de valor de atributo asociada tiene un nivel de confianza sobre cierto umbral, entonces se aceptará dicha predicción automáticamente. En el caso contrario, las predicciones de valores de atributo deberán ser validadas de forma manual a través de preguntas generadas por el sistema. Se detalla el subproceso Detectar

Valores de Atributos en la primera sección del capítulo.

En la segunda sección del capítulo, se presentan los resultados de las pruebas realizadas para evaluar el funcionamiento del subproceso para detectar valores de atributos. Se evalúan dos puntos: *(i)* el porcentaje de valores de atributos que se logra descubrir y *(ii)* el nivel de supervisión humano requerido para lograr lo anterior.

3.1. Descripción del subproceso

El subproceso comienza con el *Método Predecir Valores de Atributos*. Este incluye a su vez dos métodos, cada uno encargado de predecir los valores de un tipo de atributo en especifico (nominal o ratio) (Figura 3.2). Al momento de crear un atributo en el sistema se debe definir si es que este es nominal o ratio, por lo que el método conocerá cual de los dos sub métodos utilizar para cada atributo.

Se recuerda al lector las definiciones de ambos tipos de atributos:

> "Los atributos nominales son aquellos que toman valores discretos que no pueden ser ordenados en una escala inherente, mientras que los atributos ratio toman valores compuestos por una magnitud numérica y una unidad de medida. Los atributos ratio si pueden ser ordenados en una escala. Así, en la categoría `Notebooks`, el atributo `Memoria RAM` es ratio, tomando valores `4 gigabytes`, `8 gigabytes`, etc. mientras que el atributo `Marca` es nominal, con valores discretos como `'Lenovo'`, `'HP'`, etc.".

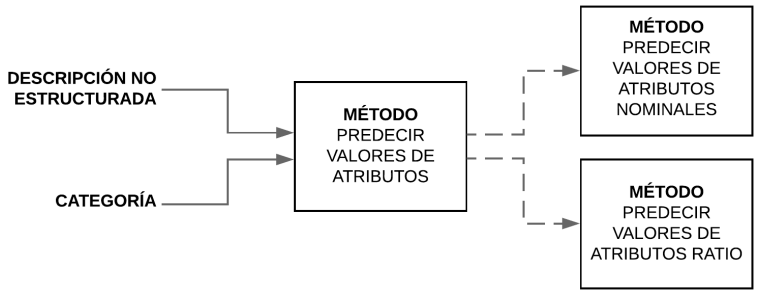

Figura 3.2: Subdivisión del método *Predecir Valores de Atributos* para ambos tipos de atributos.

3.1.1. Método Predecir Valores de Atributos Nominales

El método opera bajo el principio de que la presencia de algunos **tokens** en el nombre de un producto implican que ciertos atributos toman ciertos valores. Por ejemplo, dado que el producto con nombre `'hp(r) i5 4gb 13'' 500gb'` pertenece a la categoría `Notebooks`, el **token** `'i5'` implica que su atributo `Procesador` toma el valor `'Intel Core i5'`.

Se recuerda al lector que, en el modelo de datos, la clase **Valor de Atributo Nominal** está relacionada con la clase **Token**. Está relación puede ser interpretada como un diccionario que almacena las distintas formas de escribir un valor de atributo ratio (Figura 3.3).

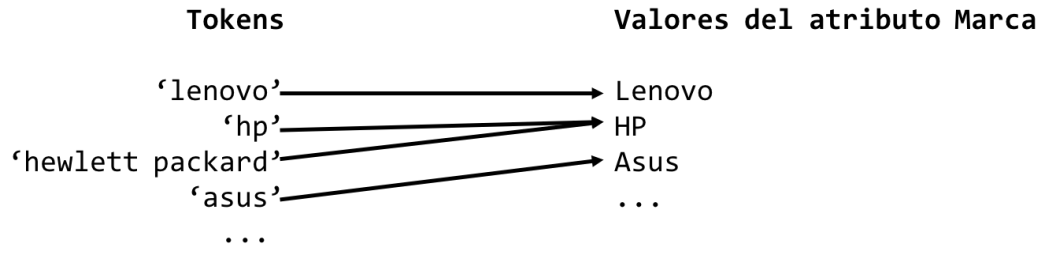

Figura 3.3: Ejemplo de valores de atributos y tokens que los implican para el atributo `Marca` de la categoría `Notebook`. Notar que un valor de atributo puede ser implicado por varios tokens: `'hp'` y `'hewlett packard'` implican al valor de atributo `'HP'`.

Se explicará el método como un proceso de 3 etapas, como se muestra en la Figura 3.4. Se procede a describir cada una de estas:

Figura 3.4: Método Predecir Valores de Atributos Nominales.

(1) Preprocesamiento

Se preprocesa el nombre del producto transformándolo a minúsculas y eliminando símbolos de puntuación. Los patrones `valor numérico + unidad de medida` también son eliminados haciendo uso de la **expresión regular** del método *Detectar Valores de Atributos Ratio* (se profundiza en esto más adelante). Se eliminan estos patrones ya que se sabe que corresponden a valores de atributos ratio y por lo tanto no serán un aporte para la detección de valores de atributos nominales.

(2) Tokenización

El nombre de producto preprocesado es dividido en fragmentos de textos, o **tokens**. A este proceso se le denomina **tokenización**. Se extraen las palabras unitarias y las frases de 2, 3 y 4 palabras, es decir los **n-grams** con n entre 1 y 4 (Figura 3.5). Se decide incluir hasta 4-grams ya que, tras inspección, se detecta que existen tokens conformados de hasta 4 palabras que implican un valor de atributo. Por ejemplo, en la categoría `Aceites` (rubro de alimentos) existe un valor del atributo `Marca` llamado `'Granjas de la Sierra'`, que es implicada por el token `'granjas de la sierra'`. Se podrían incluir **n-grams** con $n > 4$ en caso de ser necesarios. Al conjunto de token obtenidos se le conocerá como *listado de tokens*

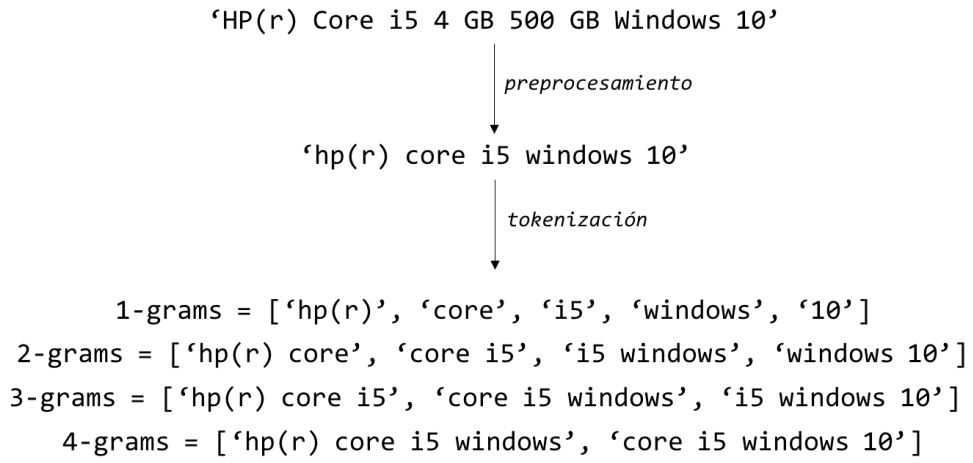

Figura 3.5: Ejemplo de preprocesamiento y tokenización para un nombre de producto

.

(3) Búsqueda en el diccionario y asignación

Sea $AN = \{an_1, an_2, ..., an_n\}$ el conjunto de atributos nominales del producto a catalogar (heredados de su categoría). Para cada token en el *listado de tokens*, se busca en el diccionario si es que este implica que un atributo de AN tome un determinado valor. Terminado el proceso de búsqueda se obtiene de resultado una tabla como la presentada en la Figura 3.6. Cada fila de la tabla representa una predicción.

NOMBRE DE PRODUCTO
'Notebook Hewlett Packard i5 i7 8GB 1 TB W10 Pro'

LISTADO DE TOKENS
['notebook', 'hewlett', ... , **'i5'**, **'i7'** ... , **'notebook hewlett'**, 'hewlett packard', ... , **'w10 pro'**, 'notebook hewlett packard', ... , 'notebook hewlett packard i5', ...]

RESULTADOS BUSQUEDA DICCIONARIO

orden	token	atributo nominal	valor de atributo nominal	confianza
1	'w10 pro'	Sistema Operativo	'Windows 10 Pro'	1
2	'hewlett packard'	Marca	'HP'	1
3	'i7'	Procesador	'Intel Core i7'	0.5
4	'i5'	Procesador	'Intel Core i5'	0.5

Figura 3.6: Resultado del método Predecir Valores de Atributos Nominales para un producto ejemplo

La búsqueda de los tokens (del listado de tokens) en el diccionario se diseña optimizando la carga computacional: comienza con los tokens de mayor número de palabras (4-grams) del listado y termina con los tokens de solo una palabra (1-grams). En el caso de que algún **n-gram** con $n > 1$ exista en el diccionario, se eliminarán los tokens de menor tamaño contenidos en este del listado de tokens. Por ejemplo, al buscar el token `'hewlett packard'`

en el diccionario se encuentra que este implica que el atributo nominal `Marca` toma el valor `'HP'`; luego los los tokens `'hewlett'` y `'packard'` serán eliminados del listado de tokens.

3.1.2. Métrica de confianza

La métrica de confianza de la predicción depende del número de valores de atributos distintos detectados para un mismo atributo nominal. Por ejemplo, analizando el caso presentado en la Figura 3.6, existen dos tokens, `'i5'` e `'i7'`, que implican que el atributo nominal `'Procesador'` debe tomar dos valores distintos (`'Intel Core i5'` e `'Intel Core i7'`). En ese caso se añaden ambas predicciones, cada una con confianza 1/2. Formalizando lo anterior, la métrica de confianza para una predicción será igual a $1/n$, con n el número de predicciones de valores de atributo distintas para el atributo en cuestión.

3.1.3. Método Predecir Valores de Atributos Ratio

Se explica el funcionamiento del método como un proceso de 3 etapas (Figura 3.7), las cuales son presentadas a continuación:

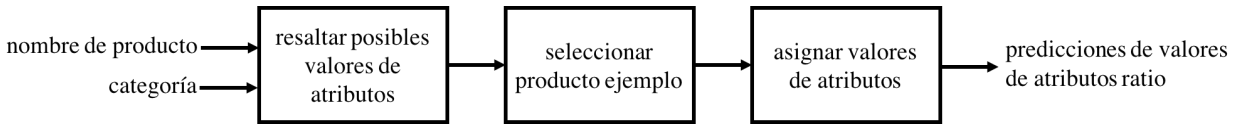

Figura 3.7: Método Predecir Valores de Atributos Ratio.

(1) Resaltar posibles valores de atributos

Haciendo uso de una **expresión regular**, se detectan todos los patrones `[valor numérico + unidad]` presentes en un nombre de producto. A los patrones detectados se les llamará *posibles valores de atributos ratio*. Se utiliza la información de los sinónimos y las abreviaciones de las unidades almacenadas en la base de datos del sistema para hacer más robusta la detección. Por ejemplo, los patrones `'500 cc'` y `'500 ml'` serán detectados como el posible valor de atributo `500 [mililitro]`[1] (Figura 3.8).

Tras esta etapa, se conocen los *posibles valores de atributos ratio* que aparecen en el nombre del producto, pero se desconoce a que atributo pertenece cada uno. A modo de ejemplo, al resaltar los *posibles valores de atributos ratio* en el producto `'Notebook Lenovo 8 GB 500 GB'` se obtiene que existen dos posibles valores `'8 [gigabytes]'` y `'500 [gigabytes]'`, pero se desconoce que el primero corresponde al atributo `'Memoria RAM'` y el segundo al atributo `'Disco Duro'`.

[1]Cuando la unidad se presente entre corchetes, se hace referencia al objeto de la clase unidad

```
PRODUCTO: 'Pack Cerveza Corona Extra 6 botellas 355 cc'
PATRONES DETECTADOS:
  (355.0 mililitro , [volumen], 0.355 litro)
  (6.0 unidad , [cantidad], 6.0 unidad)

PRODUCTO: '1x4"x3,20m Pino cepillado seco Genérico'
PATRONES DETECTADOS:
  (3.2 metro , [longitud], 3.2 metro)
  (1.0 pulgada , [longitud], 0.0254 metro)
  (4.0 pulgada , [longitud], 0.1016 metro)

PRODUCTO: 'Notebook Asus 1 TB 8GBS 14"'
PATRONES DETECTADOS:
  (14.0 pulgada , [longitud], 0.35559999999999997 metro)
  (8.0 gigabyte , [tamanodato], 8000000000.0 byte)
  (1.0 terabyte , [tamanodato], 1000000000000.0 byte)
```

Figura 3.8: Ejemplos de los posibles valores de atributos detectados por medio de la expresión regular y la información que se puede rescatar de cada uno de estos utilizando el modelo de datos. El primer elemento de la tupla corresponde al valor en la unidad detectada, el segundo elemento a la dimensionalidad de la unidad y el tercer elemento corresponde a la transformación a unidad base.

(2) Seleccionar producto ejemplo

De los productos consolidados en la categoría del producto que se está catalogando, se seleccionan los productos que tienen todos los valores de sus atributos ratio definidos. De estos últimos, se selecciona aquel con precio más cercano al producto a catalogar. Este producto será conocido como *producto ejemplo*. En la etapa siguiente se compararan los atributos ratio del producto a catalogar con el producto ejemplo. Es por esto que se selecciona como producto ejemplo al más cercano en precio, bajo el supuesto de que productos con precios similares tendrán atributos ratio similares, y por lo tanto la comparación entregará mejores resultados.

El método Predecir Valores de Atributos Ratio requiere la existencia de un producto ejemplo, por lo que para ser ejecutado en un producto (producto a catalogar), se necesita que exista al menos un producto en la categoría del producto a catalogar con todos los valores de sus atributos ratio definidos

(3) Asignar valores de atributos

Sea $AR = \{ar_1, ar_2, ..., ar_m\}$ el conjunto de atributos ratio del producto a catalogar (heredados de su categoría) y sea el $VR = \{vr_1, vr_2, ..., vr_n\}$ el conjunto de los *posibles valores de atributos ratio* detectados en la etapa *(1)*. El problema que se debe resolver a continuación

es el de asignar los n posibles valores a los m atributos ratio del producto[2].

El problema de asignación puede ser dividido por dimensionalidad. Supongamos un producto de la categoría `Notebooks` con 3 atributos ratio: `{Memoria RAM, Disco Duro, Tamaño de Pantalla}` y 3 posibles valores de atributo ratio: `{4 [gigabytes], 750 [gigabytes], 13 [pulgadas]}`. Los dos primeros atributos pertenecen a la dimensionalidad `tamaño de datos`, por lo que no pueden tomar un valor de atributo cuya unidad pertenezca a otra dimensionalidad (como `[pulgada]`). De la misma forma, el atributo `Tamaño de Pantalla` no puede tomar valores de atributo con unidad `[gigabyte]`. Así, el problema de asignación puede ser dividido en dos: *(i)* asignar los posibles valores de atributo `{4 [gigabytes], 750 [gigabytes]}` a los atributos ratio `{Memoria RAM, Disco Duro}` y *(ii)* asignar el posible valor de atributo `{13 [pulgadas]}` al atributo `{Tamaño de Pantalla}` (trivial).

Formalizando lo anterior, el problema de asignación será dividido en tantos subproblemas como dimensionalidades existan en el conjunto A. Se denotará A_{d} al conjunto de atributos ratio pertenecientes a la dimensionalidad d y, m_{d} al tamaño de este conjunto ($m_{\mathrm{d}} = |A_{\mathrm{d}}|$). De la misma forma, V_{d} es el conjunto de los posibles valores de atributos ratio de la dimensionalidad d y n_{d} el tamaño de este conjunto ($n_{\mathrm{d}} = |V_{\mathrm{d}}|$).

Una asignación válida debe satisfacer que:

- cada posible valor de atributo debe ser asignado a lo más atributo a un atributo.
- cada atributo debe tener a lo más un posible valor asignado.
- el número de atributos con un posible valor asignado debe ser igual a mín$\{n_{\mathrm{d}}, m_{\mathrm{d}}\}$.

Se calcula una métrica a cada una de las asignaciones validas en base a los valores de atributo ratio que presenta el producto ejemplo. La métrica que se utiliza es la suma de las distancias euclidianas[3] entre cada uno de los posibles valores de atributos asignados y el valor que presenta el producto ejemplo en dicho atributo. Finalmente se elige la asignación con menor distancia total como la asignación correcta (Figura 3.9).

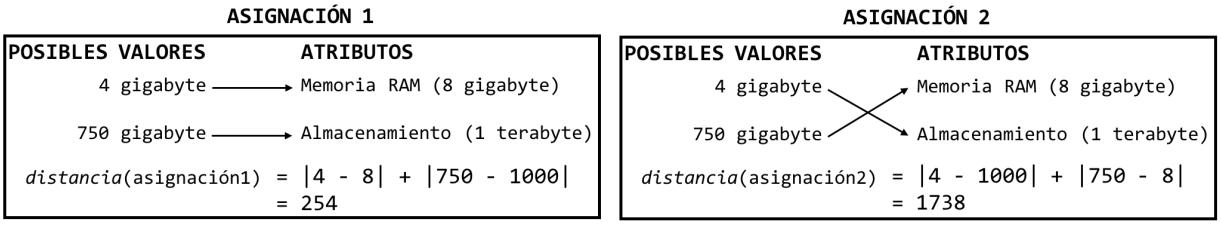

Figura 3.9: Ejemplo de posibles asignaciones y sus respectivas distancias para el producto con nombre `'HP 340 g2 i5 14.0"4GB 750GB'`, clasificado en la categoría `Notebooks`. Se muestra el subproblema de asignación para la dimensionalidad `tamaño de datos`. La asignación 1 es la que posee menor distancia al producto ejemplo y, por lo tanto, es la elegida como correcta.

[2]Notar que no necesariamente se cumple que $n = m$ (puede ocurrir tanto $n > m$ como $n < m$).

[3]Se podría utilizar otro tipo de distancia como métrica. Se decide utilizar la norma euclidiana por su simpleza y los buenos resultados que entrega en la evaluación.

Métrica de confianza

A los valores de atributos entregados por la asignación correcta (i.e. la asignación con menor distancia total) se les asigna una métrica de confianza binaria, es decir, que toma valores 0 o 1. Un valor de atributo tendrá confianza igual a 1 solo si es está dentro del rango de valores de atributos consolidados para dicho atributo. En el caso contrario, se ingresará el valor con confianza igual a 0. Los valores de atributos con métrica de confianza igual a 1 serán aceptados de forma automática mientras que los con confianza igual a 0 serán enviados a validación manual.

A modo de ejemplo, si existe solo un producto catalogado en la categoría `Notebooks`, y este producto tiene atributo ratio `Memoria RAM` con valor `'8 [gigabytes]'`, entonces cualquier posible valor de atributo distinto a `'8 [gigabytes]'` será ingresado al sistema con confianza 0, ya que no estará dentro del "rango" (en estricto rigor, en este caso no existe un rango, ya que solo hay un valor consolidado). Si posteriormente se valida `'2 [gigabytes]'` como valor de atributo (para `Memoria RAM`), entonces se validara automáticamente cualquier posible valor entre `'2 [gigabytes]'` y `'8 [gigabytes]'` (e.g. `'4 gigabytes'`).

3.1.4. Validación de Predicciones

.

Predicciones de valores de atributo nominal

Las predicciones de valor de atributo nominal son aceptadas de forma automática si su métrica de confianza es igual a 1. Para cualquier valor menor a 1, la predicción será validada manualmente por una persona a través de una pregunta de selección múltiple generada por el sistema. La pregunta en este caso es de dos etapas, primero se pregunta a que atributo corresponde el token y luego si es que este implica un valor de atributo conocido o nuevo (Figura 3.10). Los tokens que se presentan como alternativas en la primera etapa son llamados *tokens relevantes* y corresponden a un subconjunto del listado de tokens del producto.

Los *tokens relevantes* consisten en todos los n-grams del listado de tokens que se aparezcan en al menos 2 productos del catálogo. Los n-grams con $n \geq 2$ además deben satisfacer que la *métrica de selección de n-grams* supere cierto valor de umbral:

La métrica de selección se basa en la frecuencia con la que aparece el n-gram en relación a la frecuencia con la que aparecen los 1-gram que lo conforman por separado.

- Consideramos primero el caso $n = 2$, en este caso la métrica es:

$$P(x_1x_2) = \frac{count(x_1x_2)(1 - \frac{1}{count(x_1x_2)+\delta})}{count(x_1) + count(x_2) - count(x_1x_2)} \tag{3.1}$$

  en donde $count(x_1x_2)$ corresponde al número de veces que las palabras x_1 y x_2 aparecen consecutivamente (y en este orden). El paréntesis del numerador es un factor entre 0 y

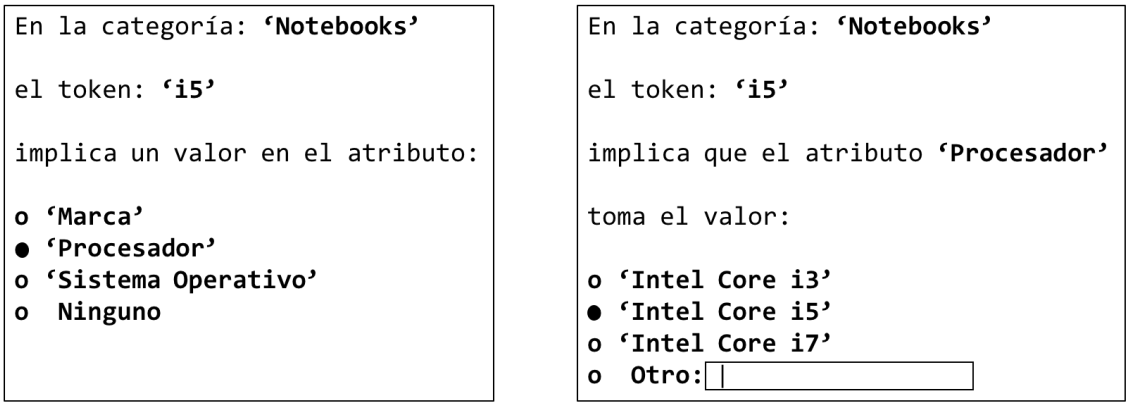

Figura 3.10: Ejemplo de preguntas de validación manual en dos etapas para las predicciones de valores de atributo nominal.

1 que se coloca para disminuir la probabilidad cuando el par de palabras aparece muy pocas veces (i.e. cuando $count(x_1x_2)$ es bajo). δ corresponde a un parámetro que ajusta el nivel de la penalización. Para este trabajo se utiliza $\delta = 2$.

- Esta métrica puede ser generalizada para todo $n \geq 2$ mediante por medio de la siguiente ecuación: 3.2)

$$P(x_1x_2...x_n) = \frac{count(x_1x_2...x_n)(1 - \frac{1}{count(x_1x_2x_n...)+\delta})}{count(x_1 \cup x_2 \cup ...x_n)} \tag{3.2}$$

En la Figura 3.11 se presentan como ejemplos los 2 grams con mayor nivel de confianza para dos categorías. Se decide fijar un umbral de 0.7 para los n-grams con $n \geq 2$ y descartar aquellos que tengan un valor menor al umbral.

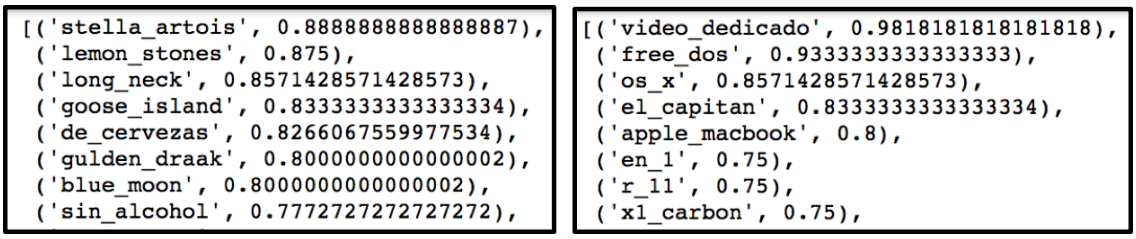

Figura 3.11: 2-grams con mayor métrica para las categorías `‘Cervezas’` (izquierda) y `‘Notebooks’` (derecha).

Predicciones de valores de atributo ratio

Si la métrica de confianza asociada a una predicción de valor de atributo ratio es menor a 1, entonces deberá ser validada de forma manual. Al igual que los casos anteriores, la validación manual consiste en responder manualmente una pregunta de selección múltiple. En este caso, las alternativas serán 2: `Si` y `No`. Si no se conoce la respuesta a la pregunta, se permitirá dejarla en blanco. En la Figura 3.12 se presenta un ejemplo de la validación para las predicciones de valores de atributo ratio

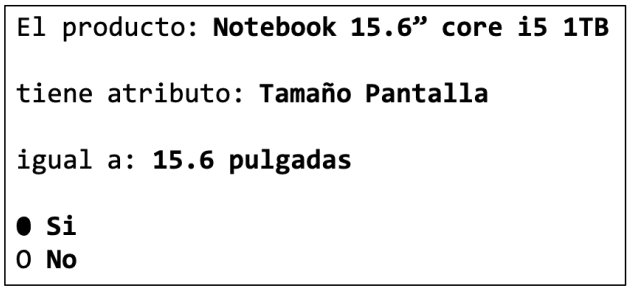

Figura 3.12: Ejemplo de pregunta de validación manual para las predicciones de valores de atributo ratio.

3.2. Pruebas para validar el subproceso

Son dos los puntos que se analizarán para determinar la eficiencia del subproceso y los Métodos que lo componen. El primer punto, es el porcentaje de valores de atributos de los productos que se logran descubrir automáticamente utilizando los Métodos Predecir Valores de Atributos Nominales y Predecir Valores de Atributos Ratio. El segundo punto, corresponde al nivel de supervisión humana que se necesita para lograr lo anterior. El subproceso será mejor en cuanto detecte una mayor cantidad de valores de atributo requiriendo un menor grado de supervisión manual.

Para analizar el comportamiento del subproceso, con respecto a los dos parámetros indicados anteriormente, se realizan pruebas para una categoría en especifico: `Notebooks`. Se define la categoría `Notebooks` en el sistema con 3 atributos ratio: `Disco Duro`, `Memoria RAM` y `Tamaño de Pantalla`; y 3 atributos nominales: `Marca`, `Procesador` y `Sistema Operativo`.

La prueba a realizar consiste en utilizar el subproceso Detectar Valores de Atributos para encontrar de los valores de los atributos de un conjunto de productos de prueba. Como productos de prueba, se utilizan datos de 264 productos (`Notebooks`), provenientes de 4 tiendas e-commerce nacionales obtenidos mediante **web scraping**. Para cada producto se cuenta con su nombre de producto, su precio y sus valores de atributos (etiquetados manualmente). En el Apéndice C, se presenta el detalle de los datos utilizados. Para poder ejecutar el Método Detectar Atributos Ratio es necesario determinar los valores de atributos ratio de al menos 1 producto de la categoría, por lo que se selecciona y definen los valores de atributos ratio de un producto en el sistema (seleccionado de forma aleatoria).

3.2.1. Resultados y discusión

Los resultados obtenidos se presentan de forma gráfica en las Figuras 3.13 y 3.14. Se puede desprender de los resultados que el Método Detectar Valores de Atributos Ratio tiene un excelente desempeño, logrando detectar correctamente los valores de los 3 atributos ratio para más del 90 % de los productos, utilizando solo 7 preguntas de validación manual. Aprovechar la estructura `[valor numérico + unidad de medida]` para estos atributos permite que su detección de nuevos valores sea más expedita. Por otro lado, los atributos nominales son

descubiertos de forma mucho más lenta y se requiere un gran número de preguntas (alrededor de 220) para poder determinar los valores de atributos de los productos. Solo se detecta correctamente los valores para alrededor de un un 80 % de los productos. Este método tiene un amplio espacio para ser mejorado y será parte del trabajo futuro a realizar en el sistema. Se debe diseñar un método Predecir Valores de Atributos Nominales que agilice la detección de nuevos valores de atributos nominales (en terminos de la supervisión humana necesaria).

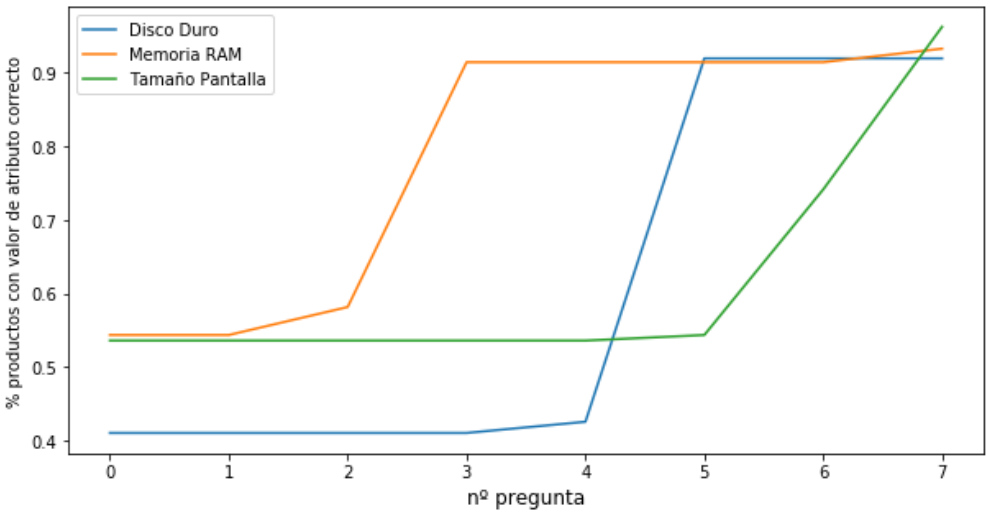

Figura 3.13: (Atributos Ratio) Porcentaje de productos con su valor de atributo detectado respecto al número de preguntas respondidas.

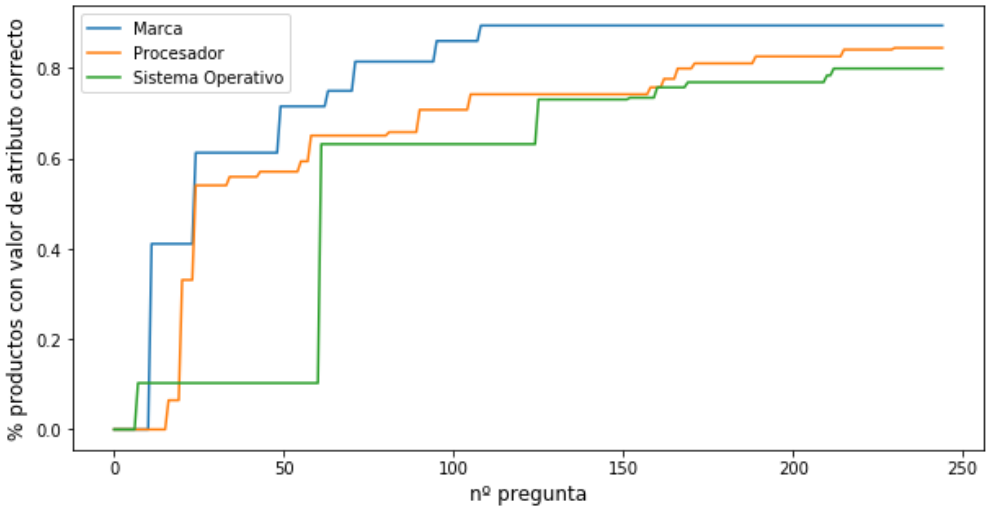

Figura 3.14: (Atributos Nominales) Porcentaje de productos con su valor de atributo detectado respecto al número de preguntas respondidas.

Capítulo 4

Prototipo del sistema y manual de uso

En el siguiente capítulo se describen brevemente los aspectos técnicos de la implementación del prototipo del *sistema de catalogación*. Durante el capítulo, cada vez que se nombre el *sistema de catalogación* se estará haciendo referencia a la versión prototipo desarrollada para este proyecto de tesis. Se presenta también un manual de uso del prototipo. En este manual se explica paso a paso como se construye un catálogo de productos en el sistema, y como se pueden aprovechar las herramientas de automatización que este presenta. El manual está dirigido a un público que no necesariamente posee conocimientos técnicos en matemáticas, algoritmos o programación, por lo que se ocupará un lenguaje cotidiano y no se profundizará más de lo necesario en el funcionamiento del sistema. El manual de uso corresponde a una sección auto contenida, por lo que si el lector ha leído los capítulos anteriores del presente informe, puede encontrar información redundante.

4.1. Prototipo del sistema de catalogación

El prototipo del sistema de catalogación está programado Python, haciendo uso del framework Django. El desarrollo del prototipo está focalizado en en el funcionamiento de la lógica del sistema, es decir, en el backend. Fue desarrollado principalmente para mostrar de forma gráfica el funcionamiento del sistema y también para realizar pruebas de su funcionamiento. El prototipo actualmente se encuentra operando de forma local y se está trabajando en su despliegue a la web (deployment).

4.2. Manual de Uso

Se explicará como construir un catálogo de productos en el *sistema de catalogación* y como utilizar las herramientas de automatización que este presenta para facilitar el trabajo. Se comenzará con un catálogo vacío (sin productos) y se procederá a agregar elementos al sistema.

4.2.1. Vocabulario

Es necesario definir ciertos conceptos que están presentes en el *sistema de catalogación* Un *producto* corresponde a cualquier artículo que puede ser comercializado. El conjunto de todos los productos presentes en el sistema será denominado *catálogo de productos* o simplemente *catálogo*. Un *atributo* corresponde a una característica que describe cierta propiedad de un producto. `Marca`, `Color` y `Peso` son ejemplos típicos de atributos. Un *valor de atributo* corresponde al valor específico que toma un atributo. Por ejemplo, en el producto con nombre `'Notebook Lenovo 14'' Core i5 8GB 1TB W10 PRO'` el atributo `Marca` toma el valor `'Lenovo'`, mientras que el atributo `Tamaño de Pantalla` toma el valor `14 pulgadas`. Por último, una *categoría* corresponde a una agrupación de un tipo particular de productos que poseen los mismos atributos. Algunos ejemplos de categorías son `Notebooks`, `Pendrives` y `Televisores`.

4.2.2. Menú de navegación

El sistema de catalogación cuenta con un menú al costado izquierdo de la pantalla (Figura 4.1). Este menú permite al usuario navegar por las distintas secciones del sistema y se mantiene presente en la pantalla durante todo momento. A lo largo del manual, se hará mención a este mediante los terminos *menú lateral* o sencillamente *menú*.

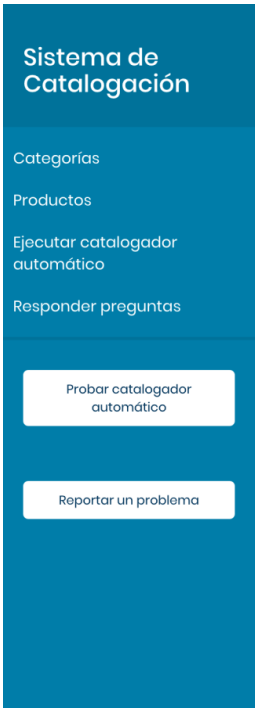

Figura 4.1: Menú de navegación

4.2.3. Añadir categorías

Al iniciar el sistema, este se encontrará vacío. Esto quiere decir que no existirán ni productos ni categorías en el sistema. Se procede a añadir dos categorías a modo de ejemplo: `SMARTPHONES` y `NOTEBOOKS`.

> **Añadir categoría:** Para añadir una categoría se debe seleccionar la sección *Categorías* del menú lateral. En esta sección se muestra un listado de las categorías que existen en el sistema. Como el sistema se encuentra vacío, no se mostrará ninguna categoría. En la parte superior se presenta un botón que permite añadir una categoría. Al hacer click sobre *Añadir Categoría* se desplegará una ventana en la que se debe ingresar el nombre de la categoría. Si se intenta crear una categoría con nombre igual a otra existente, el sistema arrojará un error y no permitirá su creación. Toda categoría es almacenada con su nombre en mayúsculas (independiente de si este fue ingresado en minúsculas).

Se recomienda definir las categorías teniendo en cuenta que todos los productos de una categoría tendrán los mismos atributos. Por lo anterior, no hace sentido definir categorías como conjuntos de productos con distintos atributos. Por ejemplo, no se deben crear categorías del estilo `NOTEBOOKS E IMPRESORAS`, ya que corresponden a una agrupación de distintos tipos de productos, con distintos atributos. Lo correcto sería definir dos categorías: `NOTEBOOKS`, con atributos `MEMORIA RAM`, `TAMAÑO DE PANTALLA`, etc.; e `IMPRESORAS` con atributos como `TECNOLOGÍA DE IMPRESIÓN`, `BLANCO Y NEGRO O COLOR`, etc. También es importante definir suficientes categorías para cubrir el universo de productos del catalogo. Un producto pertenecerá a una de las categorías definidas o no pertenecerá a ninguna. Un producto que no pertenece a ninguna categoría, tampoco tendrá atributos (ni valores de atributos) y, por lo tanto, no será posible contar con información estructurada del producto.

Luego de añadir las categorías anteriormente mencionadas, la sección *Categorías* mostrará una lista con estas dos categorías en ella (Figura 4.2). Las categorías se presentan acompañadas de 3 métricas, las cuales serán explicadas más adelante. Debajo de cada categoría existe un botón que permite eliminarla del catálogo. Al presionar sobre una categoría se visita la sección de *Detalle de la categoría*.

4.2.4. Añadir atributos

Se procede a visitar la sección detalle de la categoría `NOTEBOOKS` (Figura 4.3). Como la categoría fue recién creada, no cuenta con ningún atributo, ni tampoco con ningún producto consolidado (producto que pertenece a la categoría). Desde esta sección se pueden añadir nuevos atributos y/o productos consolidados a la categoría haciendo click en los respectivos botones.

Cabe notar que los atributos de la categoría están separados en dos tipos: *atributos nominales* y *atributos ratio*. Los *atributos nominales* son aquellos que toman valores discretos que no pueden ser ordenados en una escala inherente. Un ejemplo de atributo nominal para la categoría `NOTEBOOKS` es `Marca`, que toma valores como `'Lenovo'`, `'HP'`, `'Toshiba'`, etc.

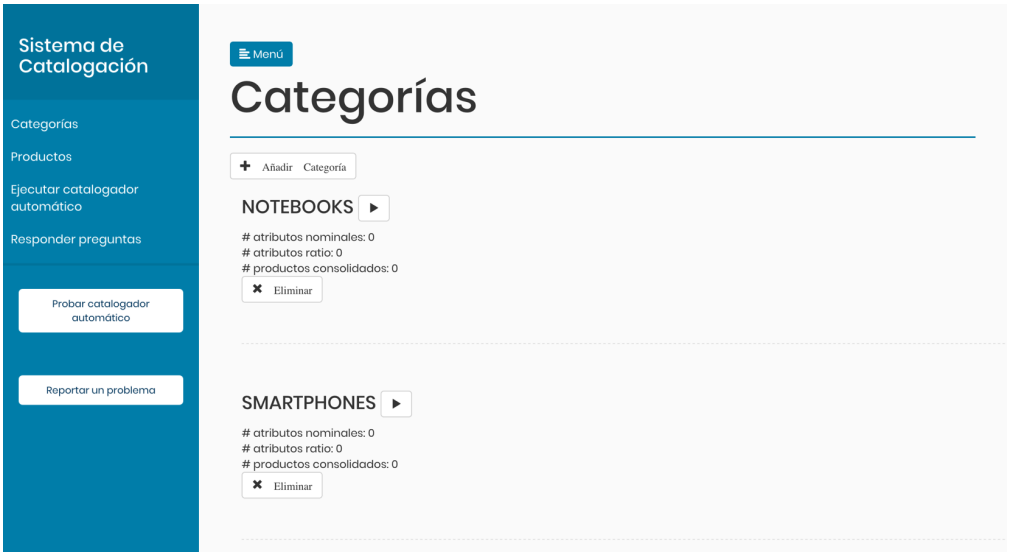

Figura 4.2: Sección `Categorías` luego de añadir las categorías `SMARTPHONES` y `NOTEBOOKS`

Por otro lado, los *atributos ratio* toma valores compuestos por una magnitud numérica y una unidad de medida. Estos últimos si pueden ser ordenados en una escala. Un ejemplo de atributo ratio, también para la categoría `NOTEBOOKS`, es `Tamaño de Pantalla`, que toma valores como `'13 pulgadas'`, `'14 pulgadas'`, `'15 pulgadas'`, etc.

Añadir atributo nominal: Para agregar un atributo nominal a una categoría se debe ingresar a la sección *Detalle de la categoría* y luego presionar el botón `Agregar Atributo Nominal`. Esto abrirá un campo de texto en el que se debe ingresar el nombre del atributo nominal que se creará. No se puede ingresar como nombre de atributo nominal el nombre de un atributo que exista previamente en la categoría (nominal o ratio). Los atributos nominales son almacenados en mayúsculas.

Añadir atributo ratio: Al igual que los atributos nominales, los atributos ratio se añaden a una categoría desde la sección *Detalle de la categoría*. En esta sección se debe presionar el botón `Agregar Atributo Ratio`, el cual despliega una ventana en la que se debe definir el nombre del nuevo atributo ratio y la dimensionalidad a la que este pertenece. La dimensionalidad de un atributo ratio corresponde al concepto que este mide. Por ejemplo, el atributo ratio `TAMAÑO DE PANTALLA` mide el largo de una pantalla, por lo que se selecciona la dimensionalidad `longitud` (Figura 4.4).

En la categoría `NOTEBOOKS`, se añaden 3 atributos nominales: `MARCA`, `PROCESADOR` y `SISTEMA OPERATIVO`; y 3 atributos ratio: `TAMAÑO DE PANTALLA`, con dimensionalidad `longitud` y `ALMACENAMIENTO` y `MEMORIA RAM`, ambos con dimensionalidad `tamaño de datos`.

Figura 4.3: Sección *Detalle de la categoría* para la categoría `NOTEBOOKS`

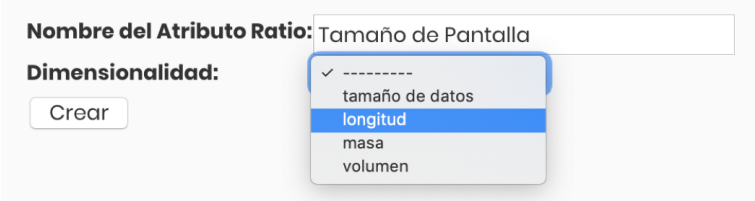

Figura 4.4: Ventana desplegada al presionar `Agregar Atributo Ratio`.

4.2.5. Añadir productos

Ahora que el catálogo ya cuenta con una categoría con algunos atributos definidos se procederá a agregar productos a esta. Existen dos formas de añadir productos al sistema. La primera consiste en añadir el producto directamente a una categoría y la segunda consiste en añadir un producto sin precisar su categoría. Ambas formas de añadir productos requieren llenar el mismo formulario; la diferencia está en como se accede a estos formularios.

Añadir productos a una categoría: Esto se realiza desde la sección *Detalle de la categoría* (de la categoría a la que se desea agregar los productos). En esta sección se debe presionar el botón *Agregar Producto Consolidado*, el cual redirige al usuario a un formulario con 2 campos: productos y precios (Figura 4.5). En el campo productos se deben ingresar los nombres de los productos que se quieran agregar, separados por un salto de línea. Los precios de los productos deben ser ingresados también separados por un salto de línea y en un formato netamente numérico (solo números). El primer precio ingresado en el campo precio será asignado al primer nombre de producto ingresado en el campo productos, el segundo precio será asignado al segundo producto y así sucesivamente.

Añadir productos sin categoría: Se debe acceder a la sección *Productos* en el menú lateral del sistema y posteriormente presionar el botón *Agregar Producto No Consolidado*. Se abrirá el mismo formulario para añadir productos presentado

anteriormente, el cual debe ser completado de la misma manera.

Figura 4.5: Formulario para añadir productos. En el ejemplo presentado en la figura se añadirán 3 productos a la categoría `NOTEBOOKS`

.

Se añaden 3 productos a la categoría `NOTEBOOKS` (Figura 4.5). Luego de añadir los productos a dicha categoría, su sección *Detalle de la Categoría* presentará una tabla con los productos que están consolidados en ella en la parte inferior de la página (Figura 4.6). En esta tabla se presentan los atributos de la categoría como columnas. Como no existen valores de atributos definidos, las celdas asociadas a estas columnas estarán en vacías

Figura 4.6: Tabla de productos consolidados en la categoría `NOTEBOOKS`

.

4.2.6. Añadir valor de atributo nominal

Como se explicó anteriormente, un valor de atributo nominal estará asociado a un atributo nominal. Por otro lado, un valor de atributo nominal tendrá asociado uno o más *tokens*. Un token corresponde a una unidad de texto que puede ser una palabra o un conjunto de palabras. Si alguno de los tokens definidos para un valor de atributo nominal está presente en el nombre de un producto, entonces se le asignará dicho valor de atributo nominal al producto.

Supongamos que se añade el valor ‘Windows 10’ al atributo nominal Sistema Operativo de la categoría NOTEBOOKS. El sistema agregará el token ‘windows 10’ de forma automática al listado de tokens asociados al valor de atributo nominal. Luego, el sistema asignará el valor de atributo ‘Windows 10’ al atributo SISTEMA OPERATIVO de todos los productos de la categoría que presenten el token ‘windows 10’ en su nombre. Existen diversas formas de escribir un valor de atributo nominal. Por ejemplo, en el producto con nombre ‘notebook dell xps 9350 13.3’’ core i5 w10’ el token que implica que el atributo SISTEMA OPERATIVO del producto toma el valor ‘Windows 10’ es ‘w10’, por lo que se deberá agregar ‘w10’ al listado de tokens asociados para poder detectar el valor de atributo en dicho producto.

Añadir valores de atributo nominal y tokens asociado: Para acceder al menú que permite añadir un valor de atributo nominal se debe visitar la sección *Detalle de la Categoría* y presionar el botón con icono de lápiz que está a la derecha de cada categoría (en la columna *Detalle*). Se redirigirá a la sección *Detalle del Atributo Nominal*. En esta sección existen dos formularios: el primero se utiliza para añadir nuevos valores de atributo y el segundo se utiliza para añadir nuevos tokens a un valor de atributo existente (se debe seleccionar el valor de atributo al cual se añadirá el token desde una lista desplegable).

Para el atributo nominal Sistema Operativo, se añaden los valores de atributo ‘Windows 10 Pro’, ‘Windows 10’ y ‘Windows 8’. Se añaden también tokens asociados a cada uno de los valores de atributo (Figura 4.7). Para eliminar un valor de atributo de un atributo nominal se debe presionar el botón con una cruz a la izquierda del valor. Para eliminar un token asociado se debe hacer click sobre este.

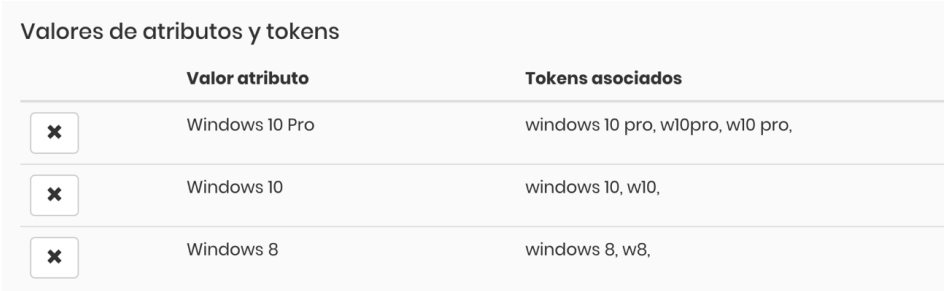

Figura 4.7: Tabla de valores de atributo y sus respectivos tokens asociados para el atributo nominal SISTEMA OPERATIVO de la categoría NOTEBOOKS

.

4.2.7. Añadir valor de atributo ratio

Un valor de atributo ratio estará compuesto por una magnitud numérica y una unidad de medida. El sistema cuenta con unidades previamente definidas. Cada unidad está definida en relación a una dimensionalidad. Por ejemplo las unidades metro, pulgada y centímetro están relacionadas a la dimensionalidad longitud.

Añadir valor de atributo ratio: Los valores de atributo ratio son añadidos

desde la sección *Detalle del Atributo Ratio*, a la cual se puede acceder desde la sección *Detalle de la Categoría* presionando el botón con icono de lápiz que se encuentra a los atributos ratio. En la sección *Detalle del Atributo Ratio* se presenta un formulario en el que se puede añadir un atributo ratio ingresando la magnitud numérica y seleccionando una unidad del listado.

Para el atributo ratio `TAMAÑO DE PANTALLA` se añaden los valores `12 pulgadas`, `13 pulgadas` y `14 pulgadas` a modo de ejemplo. La sección *Detalle del Atributo Ratio* presenta una tabla con los valores de atributo ratio consolidados (Figura 4.8), en donde se muestra el *Valor ecualizado*, que corresponde al valor en la unidad de medida en la que fue definido, y un *Valor base*, que corresponde a la magnitud numérica del atributo transformada a la unidad base de la dimensionalidad. Por ejemplo, la unidad base de la dimensionalidad `longitud` es `metro`. 1 pulgada es igual a 0.0254 metros (aproximadamente), por lo que el valor base del atributo `13 pulgadas` es $13 * 0{,}254 \approx 0{,}3048$ (Figura 4.8)

Figura 4.8: Tabla de valores de atributo para el atributo ratio `TAMAÑO DE PANTALLA` de la categoría `NOTEBOOKS`.

4.2.8. Definir los valores de atributo de un producto

El sistema permite que se ingresen manualmente los valores de atributo de un producto consolidado en una categoría. Para asignar un valor de atributo a un producto, este debe ser previamente definido en el sistema, siguiendo los pasos presentados en 4.2.6 para un atributo nominal y 4.2.7 para un atributo ratio.

Definir los valores de atributo de un producto: Se debe visitar la sección *Detalle de la Categoría* y bajar hasta donde se presenta la tabla con los productos consolidados de la categoría (Figura 4.6). Si no se alcanza a visualizar el producto buscado, se debe presionar el botón *Detalle*, que despliega la tabla completa de productos. Haciendo click a el nombre de un producto se abrirá un formulario en el que se pueden seleccionar los valores para cada atributo del producto (Figura 4.9).

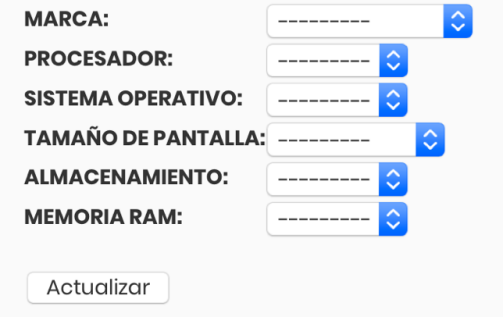

Figura 4.9: Formulario para actualizar los valores de atributos de un producto de la categoría `NOTEBOOKS`.

4.2.9. Ejecutar catalogador automático

El sistema de catalogación cuenta con algoritmos que permiten predecir la categoría y los atributos de un producto de forma automática. Estas predicciones tendrán asociado un nivel de confianza entre 0 % y 100 %. Si la confianza de una predicción es alta, entonces se aceptará esta predicción de forma automática. A modo de ejemplo, si el algoritmo que detecta categorías de producto entrega la predicción que un producto x pertenece a la categoría `NOTEBOOKS` con un nivel de confianza de 90 %, entonces se agregara este producto a la categoría de forma automática. En el caso que una predicción tenga un bajo nivel de confianza, esta deberá ser validada manualmente por el usuario a través de *preguntas de catalogación* generadas automáticamente por el sistema.

El *catalogador automático* requiere que cada categoría en el catálogo cuente con al menos 1 producto completamente catalogado, es decir, con todos sus valores de atributos definidos. Si se intenta ejecutar el *catalogador automático* sin cumplir lo anterior, el sitio arrojará un error.

> **Ejecutar catalogador automático:** Para ejecutar el catalogador automático se debe presionar el botón `Ejecutar catalogador automático` del menú lateral.

4.2.10. Responder preguntas de catalogación

Las predicciones generadas por el *catalogador automático* que no superen el umbral de confianza deberán ser revisadas de forma manual. Para estas predicciones, el sistema generará preguntas como las que se presentan en las Figura 4.10. La validación de las predicciones pendientes ocurrirá a medida que se contesten estas *preguntas de catalogación*.

> **Responder preguntas de catalogación:** Se puede acceder a las preguntas desde el menú lateral presionando el botón `Responder preguntas`. El sistema comenzará a desplegar preguntas de selección múltiples. Tras seleccionar una alternativa se debe presionar `Responder`. Si el usuario no está seguro de la respuesta de la pregunta puede omitir la pregunta presionando `Pasar`.

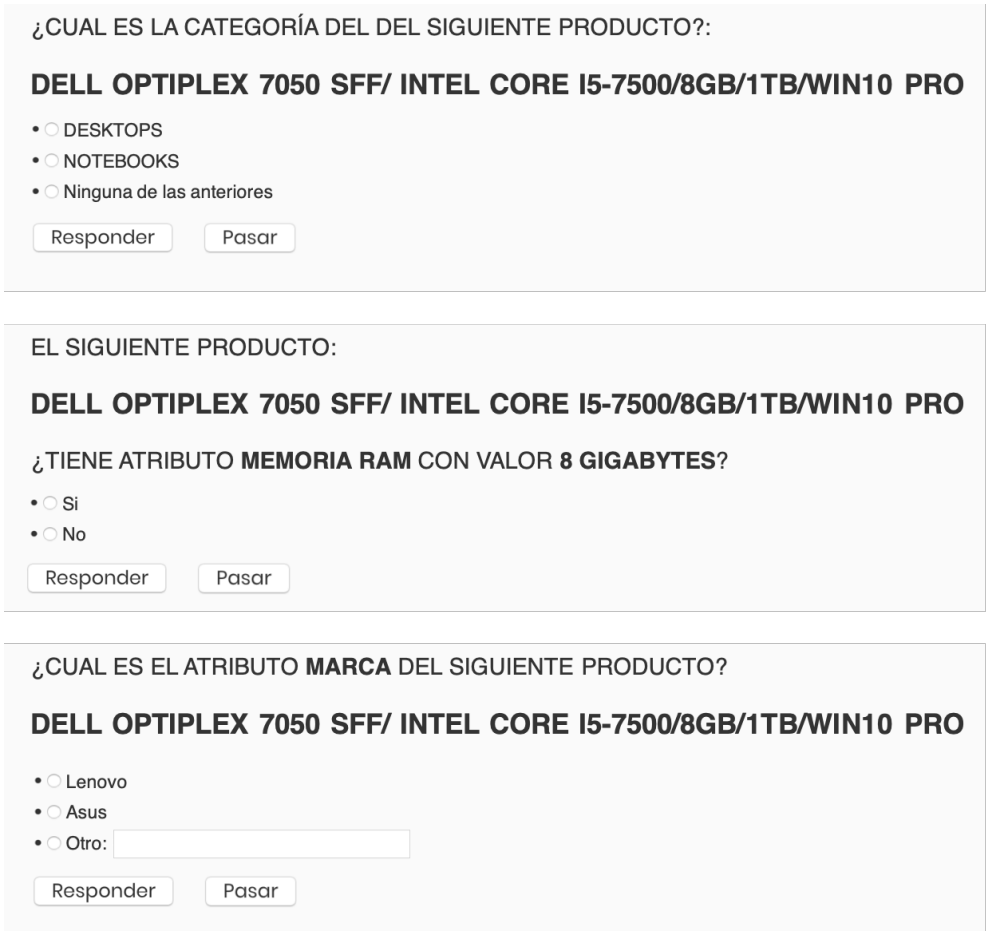

Figura 4.10: Ejemplos de *preguntas de catalogación*.

Capítulo 5

Aplicaciones

En el siguiente capítulo se presentan dos posibles aplicaciones del *sistema de catalogación* para facilitar la resolución de tareas a las que ChileCompra se enfrenta actualmente. La primera aplicación, consiste en la construcción de un comparador de precios entre productos de la tienda ChileCompra Express y alguna tienda del mercado externo. La segunda aplicación, consiste en utilizar el sistema de catalogación para construir el catálogo de productos que ChileCompra debe definir previo a construir las bases de licitación de cualquier convenio marco.

5.1. Comparador de precio

ChileCompra se encuentra constantemente comparando los precios de los productos de la tienda ChileCompra Express con el mercado externo. Si un proveedor del estado ofrece un producto en la tienda a un precio más caro que en el mercado externo, entonces esta oferta podrá ser deshabilitada de la tienda (siguiendo cierto protocolo). Para que ocurra lo anterior, la oferta debe ser un porcentaje más cara que en el mercado externo, por ejemplo, 10 %. Si se quiere realizar una comparación con el mercado externo, es necesario encontrar al menos un producto idéntico (o lo suficientemente similar) para los productos del catálogo de la tienda ChileCompra. En ChileCompra, esta labor se suele hacer de forma completamente manual, lo que consume una gran cantidad de tiempo. Esta labor puede ser facilitada por medio del sistema de catalogación. Haciendo uso del sistema, se puede estructurar la información de los atributos y los valores de atributo de los productos de forma semiautomática y, posteriormente, utilizar alguna métrica de similitud basada en los atributos y valores de atributos de los productos para identificar productos idénticos.

En esta sección se abordará un caso concreto del problema de encontrar productos idénticos en el mercado externo para comparar precios. Se procede a definir el problema concreto, para luego explicar la metodología utilizada para solucionarlo y, finalmente presentar y analizar los resultados obtenidos.

5.1.1. Caso concreto

ChileCompra necesita encontrar una comparación de precio en el mercado externo para 1713 productos del rubro de Hardware. Especificamente, estos productos pertenecen a las categorías: `Notebooks` y `Desktops`. La distribución de los productos en las categorías se presenta en la Tabla 5.1. A estos productos se les llamará *productos tienda*.

CATEGORÍA	**# PRODUCTOS**
Notebooks	679
Desktops	1034

Tabla 5.1: Distribución del número de productos por categoría en la tienda ChileCompra Express

Por otro lado, se cuenta con un listado de productos en el mercado externo. Estos datos fueron facilitados por ChileCompra y consisten en 2644 productos, también de la categoría `Notebooks` y `Desktops` (Tabla 5.2). A estos productos se les conocerá como *productos externos*.

CATEGORÍA	**# PRODUCTOS**
Notebooks	2106
Desktops	538

Tabla 5.2: Distribución del número de productos por categoría en los datos del mercado externo.

Tanto para los *productos tienda* como para los *productos externos* se cuenta con información de su categoría, su nombre de producto y su precio. En la Tabla 5.3 se presenta un ejemplo de los datos de un producto.

CATEGORÍA	**NOMBRE DE PRODUCTO**	**PRECIO**
Notebooks	HP PROBOOK 450 G3 I7 /4GB/1TB 15.6"W10 PRO	$ 561,216

Tabla 5.3: Ejemplo de datos con los que se cuenta para un producto.

5.1.2. Metodología utilizada

La metodología es la siguiente: primero, se estructura la información de los productos en un formato de tabla de atributos haciendo uso del sistema de catalogación y, posteriormente, utilizando un algoritmo simple, se calcula una métrica de similitud entre productos para todos los productos del catálogo. Si la similitud entre dos productos superan cierto umbral, entonces serán considerados idénticos.

(1) Estructurar información: Se ejecuta el sistema de catalogación vació. Se inicializa el sistema definiendo dos categorías: `Notebooks` y `Desktops`. Se definen los mismos atributos para ambas categorías. Estos son: `Marca`, `Procesador` y `Sistema Operativo`, como atributos

nominales, y `Memoría RAM`, `Almacenamiento` y `Tamaño de Pantalla`. Luego, se agregan 10 productos a cada categoría (5 de cada tienda). A estos productos, se les define manualmente todos sus valores de atributos. Posteriormente se agregan los 4337 productos (sin especificar sus valores de atributos). Se ejecuta el proceso de catalogación y se responden manualmente 10 preguntas de validación, para luego volver a ejecutar el proceso de catalogación y volver a responder 10 preguntas de validación. Se itera de esta forma hasta que el sistema no genera más preguntas. Lamentablemente no se guardó registros de este proceso, pero se estima que las preguntas respondidas fueron entre 300 y 800.

(2) Encontrar productos idénticos: La métrica de similitud entre productos utilizada es el porcentaje de valores de atributos que compartan los productos comparados (Tabla 5.4). Se calcula esta métrica entre todos los *productos tienda* y los *productos externos*. Los pares de productos cuya métrica se encuentre sobre 0.83 (5/6) serán considerados como idénticos. Si para el atributo de un producto, su valor es vació, entonces este será considerado como un atributo distinto en la comparación.

	PRODUCTO 1	**PRODUCTO 2**
MARCA	LENOVO	ASUS
PROCESADOR	INTEL CORE I3	INTEL CORE I3
SISTEMA OPERATIVO	WINDOWS 10	WINDOWS 10
ALMACENAMIENTO	1000 GIGABYTES	500 GIGABYTES
MEMORIA RAM	4 GIGABYTES	4 GIGABYTES
TAMAÑO DE PANTALLA	14 PULGADAS	15.6 PULGADAS

Tabla 5.4: Ejemplos de los atributos y valores de atributos de dos productos. La similitud entre estos es 0.5 (3/6)

5.1.3. Resultados

(1) Estructurar información: Cada producto cuenta con 6 atributos y son 4357 productos, por lo que la cantidad máxima de valores de atributos a detectar es 26142 ($4357\dot{6}$). De estos el sistema detecta 19460, es decir, un 74 % del total. Los valores de atributo no detectados pueden deberse tanto a que el sistema no logro identificar un valor de atributo presente en el nombre de producto, como a que el nombre de producto no presentaba el valor de ese atributo. Analizando los resultados de una muestra de 250 productos catalogados (seleccionados aleatoriamente) se estima que sobre el 90 % de los valores de atributo detectados son correctos.

(2) Encontrar productos idénticos: Se encuentra al menos un producto idéntico en el catálogo externo para 728 de los *productos tienda* (43 %). De los *productos tienda* que no presentan un producto idéntico en el catálogo externo, un 14 % de estos tiene menos de 5 valores de atributos definidos, por lo que no alcanzará a tener nivel de confianza superior a 5/6 independiente del conjunto de *productos externos*. Para el 86 % restante no se encuentra un producto con al menos 5 de 6 atributos con los mismos valores.

5.2. Construcción de catálogos

En ChileCompra, se debe construir *catalogo de productos* para cada *convenio marco* (CM). Un CM es definido como:

> "Una modalidad de compra de bienes y servicios a través de un catálogo electrónico o tienda virtual. (...) Cada convenio marco se asocia a uno o varios rubros o industrias y se incluye en el catálogo a través de una licitación pública que efectúa la Dirección ChileCompra."[2]

Tal como dice el extracto anterior, las licitaciones de convenio marco son diseñadas por rubros de productos. Ejemplos de CM son *CM Alimentos*, *CM Aseo*, *Linea Blanca y Menaje*, *CM Hardware*, etc.

Para licitar un convenio marco, la etapa inicial es construir un *catálogo de productos* a licitar. Esto consiste en 3 etapas:

- **Definir las categorías de productos que se licitaran.** Para el CM Alimentos algunas de estas son: *Mayonesa*, *Polvos de hornear*, *Azúcar*, etc.
- **Definir los atributos de cada categoría de productos.** En el CM Alimentos, se definen 3 atributos invariables para todas las categorías, estos son *Marca*, *Modelo* y *Medida*
- **Definir una tabla de atributos para cada producto que se licitará.** A este conjunto de productos, con información de sus tablas de atributos, se le conoce como *maestra de productos*.

Cuando el *catálogo de productos* se ha definido, se puede proceder a diseñar las bases de la licitación del CM. Aquí se decidirá, entre otras cosas, que productos deben competir entre si y cuales serán las reglas de adjudicación, que algunas veces varían entre categorías de productos.

En ChileCompra, se suele construir el *catálogo de productos* entre varias personas trabajando en paralelo sobre planillas Excel. Estas planillas posteriormente debe ser fusionadas en una versión final. Al fusionar las plantillas, suelen ocurrir los conflictos típicos de cuando varias personas trabajan en paralelo en un archivo no sincronizado. Algunos de estos son: *(i)* dos o más personas trabajaron en definir la misma información, lo que reduce la eficiencia, *(ii)* una versión de una planilla posee información distinta a otra versión, y se desconoce cual de las dos posee la información correcta, etc.

Utilizando el sistema de catalogación se puede almacenar la información del catálogo de productos en un servidor centralizado al que distintas personas puedan acceder para añadir, modificar o eliminar información. Al encontrarse sincronizada la información para toda persona que esté trabajando el catálogo, no ocurrirán ineficiencias como las definidas anteriormente,

Además, el sistema de catalogación cuenta con un proceso de catalogación capaz de automatizar el proceso de extraer tablas de atributos para los productos, por lo que ayudará a la construcción de la *maestra de producto*. La hipótesis es que se puede reducir el tiempo de

construcción de un catálogo de productos considerablemente al utilizar el sistema de catalogación. Se tenía planeado utilizar el sistema de catalogación para construir el catálogo de productos de la nueva licitación del CM Aseo. Lamentablemente, esta licitación se pospuso, por lo que no se ha podido probar concretamente la efectividad del sistema para construir catálogos de productos.

Conclusión y Trabajo Futuro

Contar con un catálogo de productos con datos estructurados de sus atributos y valores de atributos es algo cada vez más importante en el retail. A través del *sistema de catalogación*, y sus herramientas de automatización, se logra facilitar la construcción y mantención de catálogos de productos. El sistema logra automatizar en gran parte la mantención de un catálogo a través del *proceso de catalogación* y de esta forma hace factible que grandes catálogos de productos puedan ser administrados por pequeños equipos de personas.

Se cree que el principal aporte de este proyecto de tesis es la forma en la que se estructura el problema de catalogar productos automáticamente, y la arquitectura que se utiliza en el sistema para resolver este problema. Lo primero es que se divide el problema en dos etapas más simples de resolver: primero se detecta la categoría del producto (y así sus atributos), y luego se detecta los valores de los atributos del producto. Por otro lado, la arquitectura semiautomática del sistema permite minimizar los errores al catalogar productos, ya que, validara automáticamente solo las predicciones con alto nivel de confianza. El resto de las predicciones, las cuales tienen una mayor probabilidad de ser erróneas, deberán ser validadas de forma manual a través de preguntas de selección múltiple. El diseño de la validación a través de preguntas de fácil respuesta, permite que esta tarea sea externalizada a plataformas de *crowdsourcing*. Finalmente, la arquitectura del sistema se encuentra modularizada de manera tal que los métodos de predicción utilizados son fáciles de reemplazar en cualquier momento, sin necesidad de hacer mayores ajustes. Esto permite que las mejoras al sistema sean más rápidas de implementar.

La principal conclusión de este trabajo es que en ciertos casos, como el de catalogar productos, en donde es muy difícil generar un conjunto de observaciones etiquetadas lo suficientemente grande como para asegurar un buen desempeño de el(los) algoritmo(s) de predicción, un enfoque *semiautomático* puede ser una solución. Lo importante en este caso es tener en cuenta que la supervisión humana es costosa, por lo que debe estar bien enfocada. En el caso del sistema, se pueden detectar las predicciones con mayor probabilidad de ser erróneas, a través de la métrica de confianza, y así, focalizar la supervisión en estas.

Se planea orientar el trabajo futúro en 3 lineas:

- Se busca integrar el proceso de validación manual a alguna plataforma de *crowdsourcing* (propia o externa). Una plataforma de crowdsourcing es una plataforma de colaboración abierta en la una comunidad de personas (*crowd*) realiza cierta tarea a cambio de una remuneración. Se planea utilizar una plataforma de este tipo para subir las preguntas de validación manual generadas por el sistema y permitir que estas sean respondidas

por el crowd. De esta forma se puede disminuir el tiempo en el que se responden estas preguntas, al ser resueltas por un mayor grupo de personas. Es posible que el costo asociado al proceso de validación también disminuya [11].

- También se planea trabajar en mejorar los métodos de predicción utilizados en el sistema. Se tienen planificadas mejoraras para cada uno de los tres métodos –predecir categorías, predecir valores de atributos nominales y predecir valores de atributos ratio– pero, en particular, se priorizarán las mejoras sobre el método predecir valores de atributos nominales, ya que se considera que su desempeño fue deficiente al requerir un grado de supervisión por sobre el esperado. En concreto, se evaluará utilizar técnicas más sofisticadas de *named-entity recognition* para extraer valores de atributo, como las presentadas en [15, 13].
- Finalmente, se planea continuar con el desarrollo del sistema de catalogación hasta llegar a una versión de disponibilidad general. Esta versión deberá ser estable, libre de errores y con un nivel de calidad adecuado para ser utilizada por cualquier persona, sin la necesidad de tener conocimientos de programación. Como trabajo futuro también está el desplegar la versión final del sistema en un servidor con acceso a internet.

Glosario

accuracy Porcentaje de predicciones correctas del total de predicciones. 20, 23

average-word2vec (avg-w2v) Este método se basa en el modelo conocido como [14]. Utilizando w2v se obtiene vectores palabras, los cuales son sumados y ponderados para formar un vector producto. A modo de ejemplo,

$$\text{vector(`flan nestle vainilla')} = \frac{\text{vector(`flan')} + \text{vector(`nestle')} + \text{vector(`vainilla')}}{3}$$

. 20, 21

bag-of-words (BOW) Modelo utilizado para representar documentos de texto en un formato vectorial. Cada palabra (o otra unidad de texto) que esté presente en al menos uno de los documentos corresponderá a un elemento del vocabulario. Luego, un documento será representado por un vector de tamaño igual al tamaño del vocabulario, en donde cada dimensión estará asociada a una palabra. Si la palabra está presente en el documento, la dimensión correspondiente tomará el valor 1. En el caso contrario, tomará el valor 0. De esta forma, cada documento será representado por un vector de 0 y 1. Existen otras múltiples variaciones del modelo bag-of-words. 17, 20, 21

bigram Corresponde a las secuencias de 2 palabra consecutivas en una oración. Por ejemplo, la oración `'el perro ladra'` está compuesta por 2 bigrams: `'el perro'` y `'perro ladra'`. 20, 21

coeficiente de correlación de Pearson El coeficiente de correlación de Pearson es definido por la ecuación,

$$r = \frac{\sum_{\text{i}=1}^{n}(x_{\text{i}} - \overline{x})(y_{\text{i}} - \overline{y})}{\sqrt{\sum_{\text{i}=1}^{n}(x_{\text{i}} - \overline{x})^2(y_{\text{i}} - \overline{y})^2}} \tag{5.1}$$

en donde n es el número de observaciones, x_{i} e y_{i} son los valores que toma *hit* y la métrica de confianza respectivamente para el producto i. $\overline{x}$ corresponde al promedio de los valores de *hit* y $\overline{y}$ es el promedio de los valores de confianza. 22

expresión regular Es una secuencia de caracteres que definen un patrón de búsqueda en un campo de texto. Por ejemplo, la expresión regular `'[0-9]+'` busca "todas las secuencias de 1 o más números". 28, 30

k-nearest neighbors (kNN) Método no paramétrico utilizado para clasificación y regresión. Corresponde a un método supervisado, entrenado sobre un conjunto de vectores para los cuales se conoce su clase. Para clasificar una nueva observación, la cual corresponde a un vector en el mismo espacio que los vectores de entrenamiento, se buscan los k vectores de entrenamiento más cercanos a la nueva observación y se clasifica esta ultima en base a la clase mayoritaria de sus vecinos. Por ejemplo, si $k = 5$ y 4 de los vecinos de la nueva observación son de la clase A, se concluirá que la clase de la nueva observación es A. 18–21

n-gram Corresponde a las secuencias de n palabra consecutivas en una oración. Por ejemplo, la oración `'el perro ladra y salta'` está compuesta por dos 4-grams: `'el perro ladra y'` y `'perro ladra y salta'`. 28, 29

precisión Proporción de observaciones clasificadas correctamente del universo de observaciones clasificadas. 25

term frequency–inverse document frequency (tf-idf) Es un estadístico que busca reflejar que tan importante es una palabra (o otra unidad de texto) en un conjunto de documentos. Se compone de dos factores: *(i)* term frequency (tf), que captura la frecuencia con la que ocurre un termino en un documento e *(ii)* inverse document frequency (idf), que captura cuanta información entrega la palabra mediante el número de documentos en los que aparece. Luego, el factor tf-idf será la multiplicación de estos dos factores. Existen diversas formas de calcular tanto tf, como idf. En este trabajo se calcula tf como la frecuencia normalizada, es decir:

$$\mathrm{tf}_{t,\mathrm{d}} = \frac{f_{t,\mathrm{d}}}{\sum_{t \in \mathrm{d}} f_{t,\mathrm{d}}}$$

en donde $f_{t,\mathrm{d}}$ es el número de veces que el termino (palabra) t aparece en el documento d. Por otro lado, idf se calcula como es propuesto originalmente en [18]:

$$\mathrm{idf}_t = -\log \frac{|\{\mathrm{d} \in D | t \in \mathrm{d}\}|}{N}$$

en donde d representa un documento del conjunto de todos los documentos D, t corresponde a un termino (palabra) y N corresponde al número total de documentos ($|D|$). 17, 20, 21

token Unidad o fragmento de texto que puede corresponder a una palabra o a una composición de varias palabras. 27, 28

tokenización Es el acto de separar una secuencia de texto en fragmentos como palabras, frases, símbolos, etc. Por ejemplo, la oración `'el perro ladra'` puede ser tokenizada, entre otras formas, como: (`'el'`, `'perro'`, `'ladra'`), (`'el perro'`, `'ladra'`) o (`'el perro'`, `'perro ladra'`). 17, 20, 28

tradeoff Decisión tomada en una situación en la cual se debe perder cierta cualidad a cambio de otra cualidad. 23, 24

unigram Corresponde a las secuencias de 1 palabra en una oración. Por ejemplo, la oración `'el perro ladra'` está compuesta por 3 unigrams: `'el'`, `'perro'` y `'ladra'`. 17, 20, 21

web scraping Técnica de extracción de información desde sitios web, mediante la utilización de un programa o script que automatice el proceso. 19, 35, 60, 62

word2vec (w2v) W2v es un método no supervisado que, en base a una colección de textos (en este caso, colección de descripciones de productos), logra representar palabras (o otra unidad de texto) como vectores de dimensión fija. Esto lo logra mediante el uso de una red neuronal que se entrena sobre el contexto de las palabras. Lo potente de esta representación es que captura las relaciones sintácticas y semánticas de las palabras. El clásico ejemplo es que la operación *vector('king') - vector('man') + vector('woman')* resulta en una posición cuyo vector más cercano es *vector("queen")* [14]. 20, 21

Bibliografía

[1] Oxford Spanish Dictionary, 2018. Consultado el 21 de diciembre de 2018. URL: `https://www.es.oxforddictionaries.com/`.

[2] ChileCompra, 2018. Consultado el 23 de diciembre de 2018. URL: `https://www.mercadopublico.cl/Home/Contenidos/QueEsCM`.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[4] Christoph Dalitz. Reject options and confidence measures for knn classifiers. *Schriftenreihe des Fachbereichs Elektrotechnik und Informatik Hochschule Niederrhein*, 8:16–38, 2009.

[5] Belur V Dasarathy. Nearest unlike neighbor (nun): an aid to decision confidence estimation. *Optical Engineering*, 34(9):2785–2793, 1995.

[6] Michael Dinerstein, Liran Einav, Jonathan Levin, and Neel Sundaresan. Consumer price search and platform design in internet commerce. *American Economic Review*, 108(7):1820–59, 2018.

[7] Michael Droettboom. Correcting broken characters in the recognition of historical printed documents. In *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on*, pages 364–366. IEEE, 2003.

[8] Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48, 2006.

[9] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1262–1273, 2014.

[10] Christian Holst. The current state of e-commerce filtering. *Baymard Institute*, Abril 2015. Consultado el 3 de diciembre de 2018. URL: `https://www.smashingmagazine.com/2015/04/the-current-state-of-e-commerce-filtering`.

[11] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User*

interface software and technology, pages 43–52. ACM, 2011.

[12] Eyal Levy Grass. Rediseño de bases de licitación para convenios marco competitivos. caso de estudio basado en convenio marco de alimentos. 2017. URL: `http://repositorio.uchile.cl/handle/2250/147385`.

[13] Bodhisattwa Prasad Majumder, Aditya Subramanian, Abhinandan Krishnan, Shreyansh Gandhi, and Ajinkya More. Deep recurrent neural networks for product attribute extraction in ecommerce. *arXiv preprint arXiv:1803.11284*, 2018.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[15] Ajinkya More. Attribute extraction from product titles in ecommerce. *arXiv preprint arXiv:1608.04670*, 2016.

[16] David Moth. Nine tips to help improve your product filtering options. *Econsultancy*, Junio 2013. Consultado el 3 de diciembre de 2018. URL: `https://econsultancy.com/nine-tips-to-help-improve-your-product-filtering-options/`.

[17] Dan Shen, Jean-David Ruvini, and Badrul Sarwar. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 595–604. ACM, 2012.

[18] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[19] Stanley Smith Stevens et al. On the theory of scales of measurement. 1946.

[20] Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proceedings of the VLDB Endowment*, 7(13):1529–1540, 2014.

[21] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 1999.

Apéndice A

Modelo de datos completo

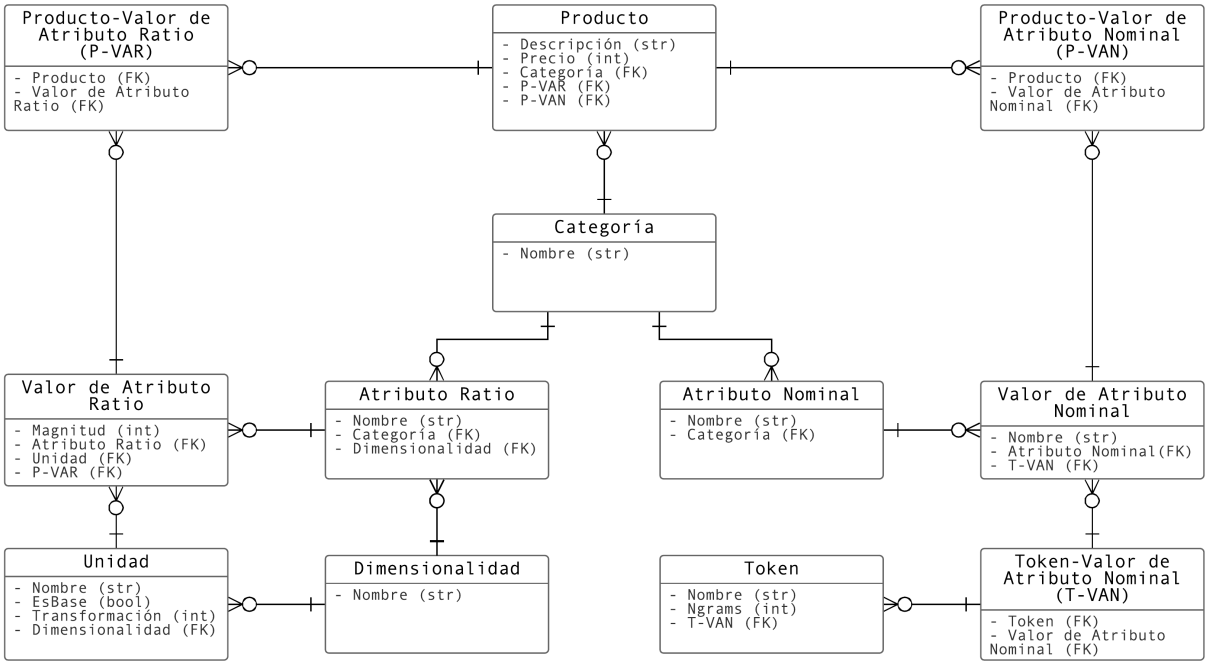

Figura A.1: Modelo de datos completo del sistema de catalogación. Los paréntesis en los atributos de las clases señalan el tipo de dato al cual corresponde: string (str), enteros (int), binarios (bool) o llaves foráneas a otra clase (FK).

Apéndice B

Datos utilizados para las pruebas realizadas en el subproceso Detectar Categoría

El set de datos utilizado para realizar pruebas está conformado por 23,181 productos provenientes de 2 tiendas e-commerce nacionales (se eliminaron 12 productos con nombres vacíos). Los datos de ambas tiendas fueron obtenidos mediante **web scraping** en sus sitios web La primera tienda corresponde a PC Factory, que comercializa productos del rubro tecnológico (e.g. notebooks, televisores). Estos datos fueron obtenidos el 28 de mayo de 2017 desde `http://www.pcfactory.cl`. La segunda tienda es Jumbo, supermercado nacional que vende productos alimenticios y para el hogar (e.g. lacteos, papel higiénico). Los datos se obtuvieron el 16 de abril del 2017 desde `http://www.jumbo.cl`. Para cada producto se cuenta con información sobre su tienda, nombre y posición en el árbol de categoría. El árbol de categorías corresponde a 3 niveles de categorías ordenados jerárquicamente. Por ejemplo el producto `'Cooper Queso Granulado 500g'` pertenece a la categoría nivel 1 `'Frescos'`, categoría nivel 2 `'Quesos'` y categoría nivel 3 `'Rallado y Granulado'`.

Se utilizan las categorías nivel 3 como categoría a predecir ya que son las más cercanas a contener solo un tipo de producto, aunque esto no siempre ocurre (e.g. `'Lápices y Marcadores'`. Se recuerda que en el sistema de catalogación las categorías estarán definidas a nivel de tipo de producto, por lo que estas categorías son las que entregarán resultados más fidedignos.

En total son 594 categorías nivel 3. Se presentan ejemplos de estas para cada tienda en la Figura B.1.

Un análisis de los datos (Figura B.2) permite comprobar que existe un desbalance en el número de productos por categoría y que la mayoria de las categorías contienen pocos productos: un 58 % de las categorías nivel 3 tienen menos de 30 productos. Este fenómeno es característico en el problema de clasificar productos de e-commerce y dificulta la capacidad predictiva de los métodos de clasificación [13].

EJEMPLOS CATEGORÍA NIVEL 3	
Jumbo	**Pc Factory**
Yoghurt Batido	Juegos XBOX One
Yoghurt Light	Juegos Playstation 4
Yoghurt Sin Lactosa	Pendrives
Lápices y Marcadores	All-in-One
Condimentos	Teclado
Maní	Smartphones
Hielo	Cables
Fruta Entera	Candados
Café Instantáneo	Notebooks
Cereales	Routers

Figura B.1: Ejemplos de categorías nivel 3 para ambas tiendas

Al analizar los datos también se descubre que existen errores en la clasificación de los producto. Una revisión manual a una muestra aleatoria de 200 productos arroja que menos de un 5 % de estos están mal clasificados. Si bien los resultados obtenidos se pueden haber visto afectados por estos errores, al ser tan bajo el porcentaje de productos con error se considera despreciables.

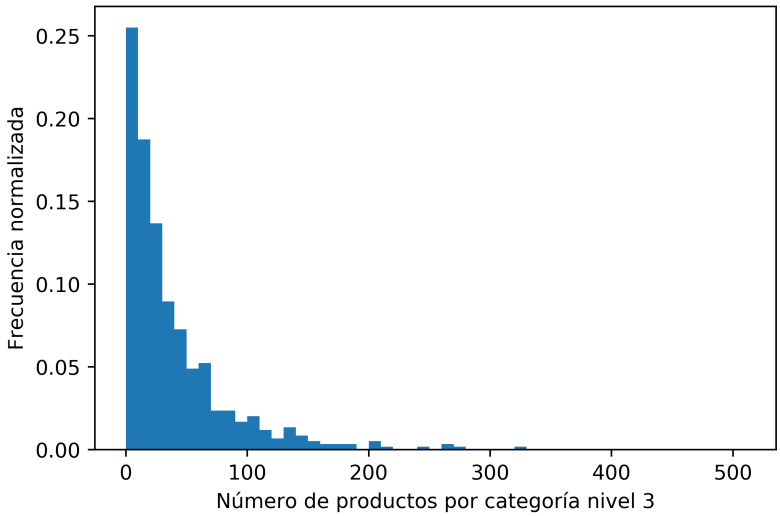

Figura B.2: Histograma cantidad de productos por categoría nivel 3

Apéndice C

Datos utilizados para las pruebas realizadas en el subproceso Detectar Valores de Atributos

El set de datos está conformado por 264 productos de la categoría `Notebooks`, obtenidos desde 4 tiendas e-commerce nacionales mediante **web scraping**. Se presenta estas tiendas, junto con los sitios desde los que se obtuvieron los datos y las fechas en las que se obtuvieron.

- **PC Factory**, cuyos datos se obtuvieron el 28 de mayo de 2017 desde `http://www.pcfactory.cl`.
- **Magens**, cuyos datos se obtuvieron el 19 de junio de 2017 desde `https://www.magens.cl/`.
- **SP Digital**, cuyos datos se obtuvieron el 24 de junio de 2017 desde `https://www.spdigital.cl`.
- **Linio**, cuyos datos se obtuvieron el 05 de agosto de 2017 desde `https://www.linio.cl`.