Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Gobierno de datos dirigido por una Inteligencia Artificial usando ontologías de dominio

Trabajo Fin de Máster

Presentado por: García Padrón, Marcos

Director/a: Tejeda Lorente, Álvaro

Ciudad: Madrid
Fecha: septiembre 2020

Resumen

Descubrir los datos que una organización posee y catalogarlos correctamente en un diccionario de términos de negocio, es un reto actual de cualquier corporación. Abordamos la solución a este problema, usando técnicas de inteligencia artificial para el descubrimiento automático de conceptos de negocio y clasificándolos con una ontología de un dominio concreto, específicamente en subdominios de *Financial Industry Business Ontology (FIBO)*, donde clasificaremos el dato técnico persistido con el concepto bancario del negocio.

Mostramos 3 técnicas para el tratamiento de datos que sirven como entrada hacia los modelos, obteniendo unos resultados de clasificación exitosos, llegando a la conclusión que la gestión de datos y, las piezas necesarias para el gobierno de los mismos, pueden enriquecerse con técnicas de inteligencia artificial aumentando así la productividad dentro de cualquier organización y facilitando el descubrimiento semántico de los datos técnicos persistidos, hacia los conceptos funcionales usando lenguajes y ontologías formales estandarizadas.

Palabras Clave: ontologías, inteligencia Artificial, gobierno de datos, lenguajes de dominio, gestión de datos, clasificación, red neuronal.

Abstract

Discovering the data that an organization possesses and cataloguing it correctly in a dictionary of business terms is a current challenge for any corporation. We address the solution to this problem, using artificial intelligence techniques for the automatic discovery of business concepts and classifying them with a specific domain ontology, specifically in Financial Industry Business Ontology (FIBO) sub-domains, where we will classify the technical data persisting with the banking concept of the business.

We show 3 techniques for data treatment that serve as input towards the models, obtaining successful classification results, reaching the conclusion that data management and, the necessary pieces for data governance, can be enriched with artificial intelligence techniques, thus increasing productivity within any organization and facilitating the semantic discovery of persistent technical data, towards functional concepts using standardized formal languages and ontologies.

Keywords: ontologies, artificial intelligence, data governance, domain languages, data management, classification, neural network.

Índice de contenidos

Índice de tablas

Índice de figuras

1 Introducción

El correcto gobierno de datos es la quimera que cualquier organización está buscando continuamente. No sólo el gobierno o control de los datos en sí mismo, sino todo el proceso interno de la organización para cumplir con estándares, leyes, reglamentos, etc. El típico ejemplo es el del "Reglamento General de Protección de Datos" o RGPD [1] donde el mero hecho de que las organizaciones estuvieran preparados para ella ha supuesto muchas iniciativas de negocio para determinadas compañías y muchos problemas para otras.

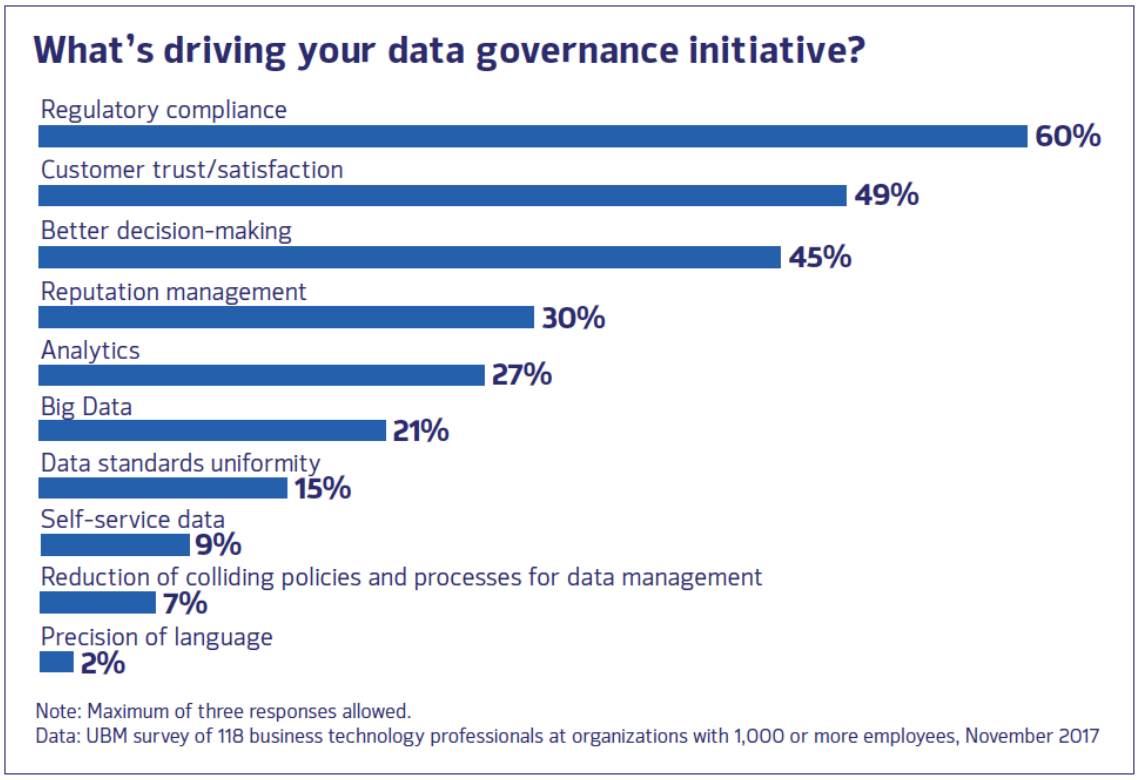


Ilustración 1- Factores que originan una iniciativa de Data Governance extraído de [49]

En este caso en concreto de la RGPD, el no saber con qué datos se cuenta en la organización, su diseño, su sensibilidad, privacidad, la seguridad asociada, su naturaleza,… en definitiva su semántica y relaciones con otros conceptos de negocio [2], ha desembocado en una inversión [3] muy grande para dar solución a algo que podría haberse evitado usando técnicas como "privacidad dirigida por el diseño", usando lenguajes de diseño de arquitectura empresarial [4] y, como pretenderemos en este trabajo, apoyándose en técnicas de inteligencia artificial [5] para conseguir una automatización en la asignación de la semántica a los datos técnicos que se encuentran en la empresa.

Algunos estándares y cuerpos de conocimiento como *Data Management Body of Knowledge (DMBoK)* [6] establecen un framework para que las compañías se puedan guiar y servir de todo lo que a la gestión de datos se refiere, y siempre, el gobierno del dato es el eje central para todas las áreas

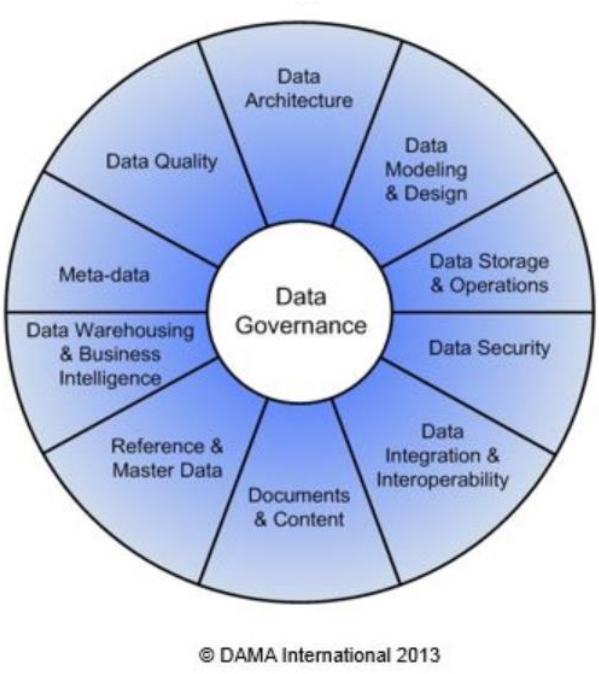


Ilustración 2 Rueda de DMBoK [6]

La creciente evolución y auge del mundo de los datos y de la inteligencia artificial, hace que cada día más se profundice en iniciativas y marcos de trabajo para homogenizar la forma en la que los datos son tratados, y por ende la forma en que desde los datos se puede extraer información y conocimiento para una mejor toma de decisiones.

1.1 Motivación

La gestión de lo que se ha conocido como "*el petróleo del siglo XXI"* es algo que siempre se ha quedado a medias y nunca se completa en las compañías. Esto es debido a que en cualquier organización siempre hay silos de información y los procesos internos nunca están optimizados. Adicionalmente, el complejo entramado organizativo que suele aparecer en las empresas, hace que la replicación y duplicidad de la información sea algo típico en las mismas, derivando esto en interpretaciones diferentes del significado de los mismos conceptos de negocio, lo que termina en el uso de un lenguaje de negocio informal y no unificado con sus consecuentes malinterpretaciones, pérdidas de tiempo y esfuerzo.

El abaratamiento del hardware y las mejoras en la computación ha hecho que el almacenamiento de los datos sea algo que se haya hecho en las organizaciones de una manera “descontrolada y desgobernada”, lo que ha derivado a tener cantidades ingentes de datos (que no información ni conocimiento como bien se diferencia en ISO 2382-36:2019 [7]).

Generalmente, este problema se suele enfocar internamente desde un punto de vista de adquirir herramientas y no como un proceso completo donde toda la organización debe de estar alineada.

1.1.1 Componentes principales en el proceso

Uno de los componentes esenciales para este gobierno de datos en una organización es tener **un glosario de términos de negocio actualizado** y vinculado con los datos técnicos que están en el lago de datos corporativo [8]. Es esta carencia que hemos identificado, lo que nos ha motivado a abordar este problema, complementando y enriqueciendo el gobierno de los datos mediante el uso de técnicas de inteligencia artificial y apoyándonos en ontologías estándares de dominios específicos [9], de tal manera que ayudemos a construirlo mediante una categorización, enriquecimiento automático y estandarizado a nivel mundial.

La principal causa para que nunca se consiga ese gobierno y correcta catalogación de los activos de datos en una organización es el desalineamiento existente entre IT y Negocio [10].

Desde un punto de vista de proceso, las oficinas del *Chief Data officer (CDO)* en las compañías, lo que pretenden es precisamente gobernar todo el proceso asociado a los datos e intentar alinear, bajo políticas y procedimientos corporativos, un framework que ayude a tener internamente una cultura dirigida por los datos [11] (*Data Driven Organisation)* y que elimine ese desalineamiento sirviendo como puente entre estos dos ‘actores’ protagonistas de las organizaciones. Sin embargo, aún teniendo políticas y procedimientos “escritos” y “publicados” en un portal corporativo, es necesario que exista una ayuda complementaria por parte de herramientas que, de una manera semi-supervisada al comienzo y no-supervisada al final, aligere los procedimientos y, para el caso en concreto que nos ocupa, que eliminen los pasos manuales que la construcción de una catálogo de conceptos de negocio conlleva, mejorando de esta manera la productividad, tiempos de descubrimiento y sobre todo el conocimiento de los activos de datos de los que disponemos de una manera completa.

Desde un punto de vista técnico, la publicación de servicios siguiendo arquitecturas *RESTFul [12]*, hace que los datos se puedan compartir de una manera totalmente libre dentro de la organización. Ahora bien, esta flexibilidad suele tener muchos problemas para los analistas y para los científicos de datos, ya que tienen que desempeñar tareas de

descubrimiento de datos de negocio de manera manual, estudiando la documentación (si la hay) e inferir semántica que quizás no es la correcta y que posteriormente tendrán que adaptar a los procesos y algoritmos, siguiendo un esquema que no tiene por qué ser el correcto ni el adecuado para el negocio que les ocupa, puesto que no se rige por ningún estándar aprobado sino por el "boca a boca" y la "prueba y error".

El hecho que los proveedores de datos de la corporación estén continuamente evolucionando y cambiando, hace que los analistas de esta información nunca puedan estar al día ya que la organización se mueve a diferentes ritmos que no convergen.

1.2 Planteamiento del trabajo

Visto todo lo anteriormente expuesto, pensamos que una solución como la que proponemos de usar Inteligencia Artificial para hacer crecer el catálogo de términos de una manera formal mediante el uso de ontologías, puede ser el germen para poder cambiar la mentalidad hacia un consumo de datos con semántica inherente y partiendo de las partículas básicas, los datos, para conseguir información y conocimiento.

Automatizar la asignación directa de los conceptos de negocio y conseguir el descubrimiento de semántica en los datos técnicos que persisten en los entornos de la organización, puede solucionar muchos de los puntos que hemos descrito anteriormente.

1.2.1 Delimitación del alcance

En primer lugar, al ser un campo demasiado amplio para abordar en un documento de estas características, acotaremos el problema a un dominio específico, que como hemos indicado anteriormente será el dominio de negocio bancario. Para ello haremos uso de una ontología estándar a nivel mundial "*Financial Industry Business Ontology" (FIBO)* [13] donde nos apoyaremos para dar semántica a los registros de un banco del sector minorista. Entrenaremos modelos preparando los datos para tal efecto, con el objetivo de actualizar y enriquecer, en tiempo real o por lotes (*batch*) dependiendo de la organización, la asignación de datos para tener correctamente un gobierno de datos consistente y actualizado con mínima intervención humana.

Aunque nos hemos restringido a un dominio concreto, cabe destacar que este problema y su resolución es perfectamente exportable a otros dominios de negocio totalmente

extensos y variopintos como por ejemplo para las organizaciones [14], para la información de sensores [15], para el sector automovilístico [16], etc... De hecho, apoyándonos en estas ontologías y vocabularios mundialmente aceptados, la aplicabilidad de una red para el mismo objetivo, actualizaría el glosario de términos de negocio de una corporación de manera automática para conseguir un proceso de gobierno de datos completo y basado en estándares internacionales de cada una de las industrias que lo soliciten.

Para representar la vista conceptual asociada a esta arquitectura concreta del dominio bancario, usamos un punto de vista de cooperación de negocio (*Business Process Cooperation)* del lenguaje de modelado de arquitectura *Archimate* [17] para dar una visión de lo que pretendemos conseguir a nivel de piezas principales de nuestra propuesta de solución.

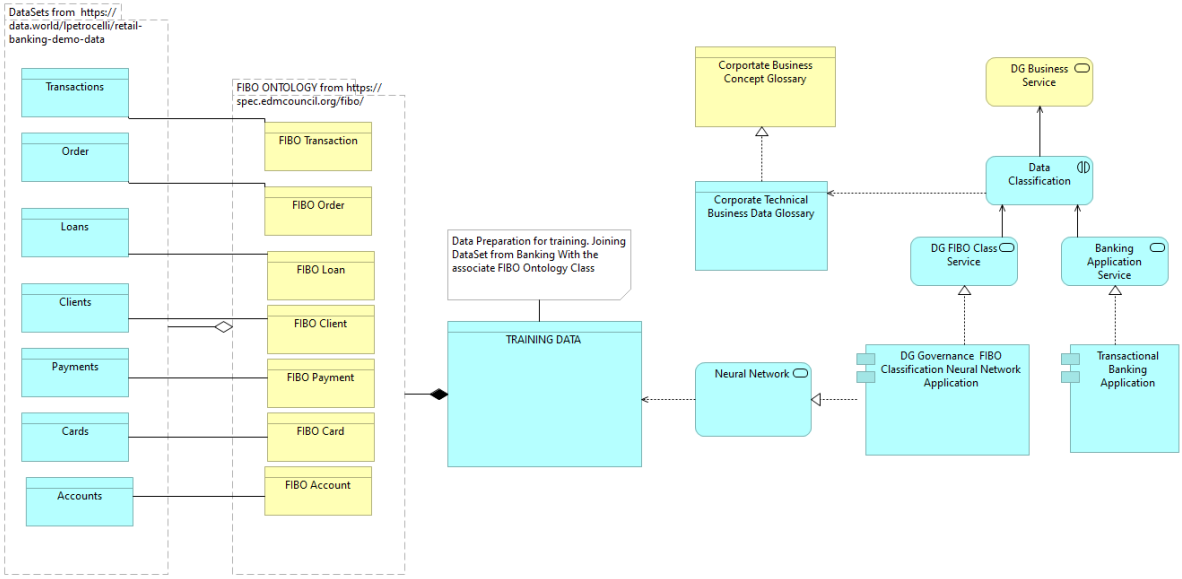


Ilustración 3 - Business Process Cooperation Architecture Viewpoint

1.3 Estructura de la memoria

Se detalla a continuación la estructura de este documento:

1. **Introducción**: se introduce el problema que se va a abordar en este TFM, junto con la motivación que lo originó y cómo se plantea realizar el trabajo.
2. **Contexto y estado del arte**: se hará referencias a investigaciones y trabajos previos sobre la temática elegida.

3. **Objetivos y metodología del trabajo**: se enumerarán y analizarán los objetivos que se persiguen. Se detallará la metodología y se justificarán las herramientas y algoritmos utilizados.
4. **Identificación de requisitos:** se identificarán las principales características y condiciones que deben presentar el conjunto de datos utilizado para construir el modelo.
5. **Descripción de la herramienta de software desarrollada**: se mostrará la aproximación de la implementación software desarrollado para acometer la solución al problema.
6. **Evaluación**: Se explicará la usabilidad y la aplicabilidad de la herramienta para resolver el problema propuesto.
7. **Conclusiones y trabajo futuro**: se presentarán las conclusiones derivadas de este TFM, así como los siguientes pasos que se podrían tomar para evolucionar el planteamiento presentado.
8. **Bibliografía**: se enumerarán las referencias bibliográficas utilizadas en este documento

2 Contexto y estado del arte

2.1 Datos y semántica

Cabe destacar que el ámbito en el que nos estamos moviendo en esta memoria es un ámbito bastante nuevo y en el que la experimentación está llevándose a cabo de una manera incipiente en estos últimos años sin, de momento, una consecuencia tácita de metas realmente logradas.

Actualmente el movimiento de la *Web semántica [18, 19, 20]* y *Open Linked data [21, 22, 23]* son corrientes totalmente vinculadas, que lo que pretenden es dotar a la web de conocimiento/inteligencia completa en base a ontologías [24]. La unión de varias ontologías [25] y la publicación de estos datos y relaciones de una manera que puedan ser consultadas y obtener conocimiento a través de sus relaciones [26, 27], es donde radica el poder de estas nuevas técnicas

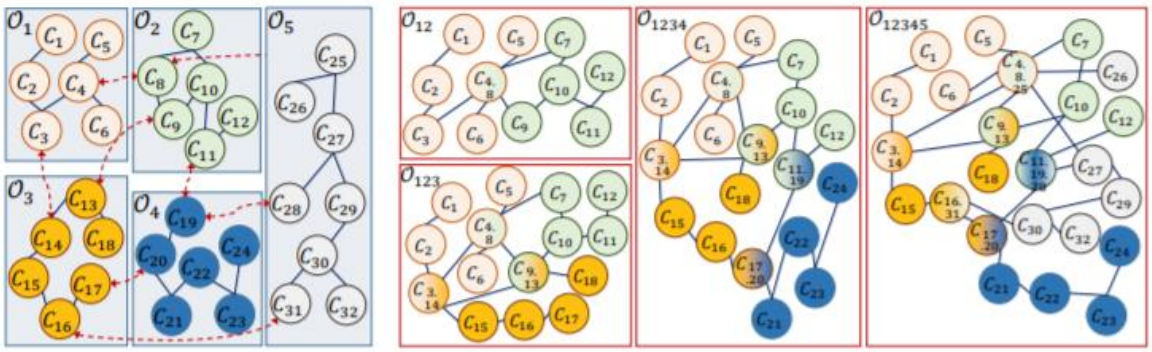


Ilustración 4 - Uniendo Ontologías – Uniendo conocimiento [25]

Las ontologías confeccionarían el universo del conocimiento de cualquier dominio y como imaginamos inter-dominio vía la relación de conceptos y clases. Estas son ideas básicas generales para plantear lo que se pretende abordar en este trabajo en una parcela específica y en una escala apropiada a su objetivo y alcance.

2.2 Iniciativas y aproximaciones actuales

Las ideas que subyacen en todas estas iniciativas y estudios, son las de publicación de datos y consultas sobre los mismos usando ontologías, pero en ningún momento se acercan a una aproximación de hacer ‘descubrimiento’ y ‘catalogación’ de su vocabulario de una manera automática.

2.2.1 Estudios:

Aproximaciones como la de Piveu [28], que lo que plantean es tener una plataforma desde una perspectiva de gestión de datos a nivel de implementación estática y directa y gestionar esos datos

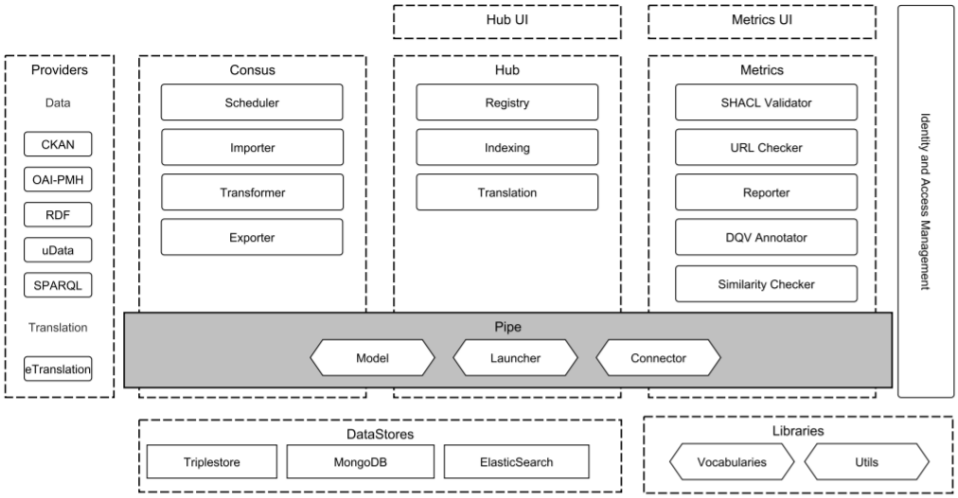


Ilustración 5 - Piveau – Arquitectura de alto nivel [28]

tampoco plantean modelos de aprendizaje y autoadaptación para solucionar el problema de categorización e inferencia de conceptos de negocio mediante datos técnicos.

Otras aproximaciones exploran el terreno de traducción directa y consultas de esos datos abiertos a través de "*SPARQL Protocol and RDF Query Language"(SPARQL)* [29] como método de consulta para inferir esa categorización y asociación de conceptos de negocio con los datos técnicos.

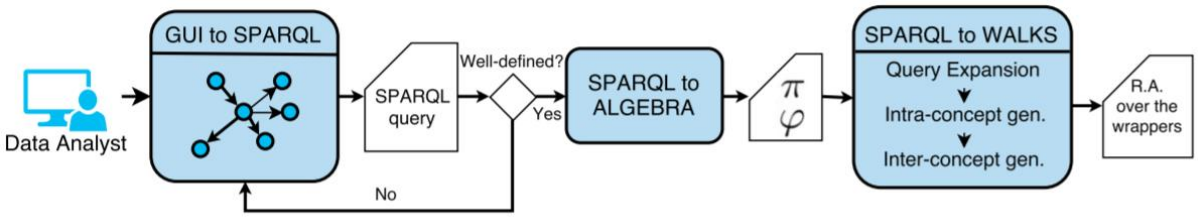


Ilustración 6 -Uso de SPARQL para asociación de conceptos[29]

Así tenemos algoritmos como los expuestos en [30]

Algorithm 1 Adapt to Release

Pre: $\mathcal{T}$ is the BDI ontology, R new release
Post: $\mathcal{T}$ is adapted w.r.t. R

function NewRelease($\mathcal{T}$, R)
  $Source_{uri}$ = "S:DataSource/"+$source(R.w)$
  if $Source_{uri} \notin$ SELECT $?ds$ FROM $\mathcal{T}$ WHERE $\langle ?ds$, "rdf:type", "S:DataSource"$\rangle$ **then**
    $\mathcal{T}.\mathcal{S} \cup= \langle Source_{uri}$, "rdf:type", "S:DataSource"$\rangle$
  end if
  $Wrapper_{uri}$ = "S:Wrapper/"+$R.w$
  $\mathcal{T}.\mathcal{S} \cup= \langle Wrapper_{uri}$, "rdf:type", "S:Wrapper"$\rangle$
  $\mathcal{T}.\mathcal{S} \cup= \langle Source_{uri}$, "S:hasWrapper", $Wrapper_{uri}\rangle$
  for each $a \in (R.w.\overline{a_{ID}} \cup R.w.\overline{a_{nID}})$ **do**
    $Attribute_{uri} = Source_{uri}$+$a$
    if $Attribute_{uri} \notin$ SELECT $?a$ FROM $\mathcal{T}$ WHERE $\langle ?a$, "rdf:type", "S:Attribute"$\rangle$ **then**
      $\mathcal{T}.\mathcal{S} \cup= \langle Attribute_{uri}$, "rdf:type", "S:Attribute"$\rangle$
    end if
    $\mathcal{T}.\mathcal{S} \cup= \langle Wrapper_{uri}$, "S:hasAttribute", $Attribute_{uri}\rangle$
  end for
  $\mathcal{T}.\mathcal{M} \cup= \langle Wrapper_{uri}$, "M:mapping", $R.G\rangle$
  for each $(a, f) \in R.F$ **do**
    $a_{uri} = Source_{uri}$+a
    f_{uri} = "G:Feature/"+f
    $\mathcal{T}.\mathcal{M} \cup= \langle a_{uri}$, "owl:sameAs", $f_{uri}\rangle$
  end for
end function

Ilustración 7 - Algoritmo de Wrapping de conceptos [30]

Tomando ventajas de *SPARQL* sobre *Resource Description Framework (RDF) [31]* y usando 'envolturas' que hacen emparejamiento directo por parte del administrador de datos (*Data Steward*), siendo éste el encargado (*humano)* de mantener viva la concordancia entre las fuentes del ecosistema de datos con las clases de la ontología que le daría la semántica relacionada y los conceptos de negocio dispuestos para ser consumidos por los analistas, científicos de datos, usuarios de negocio y, en general, la organización en su totalidad.

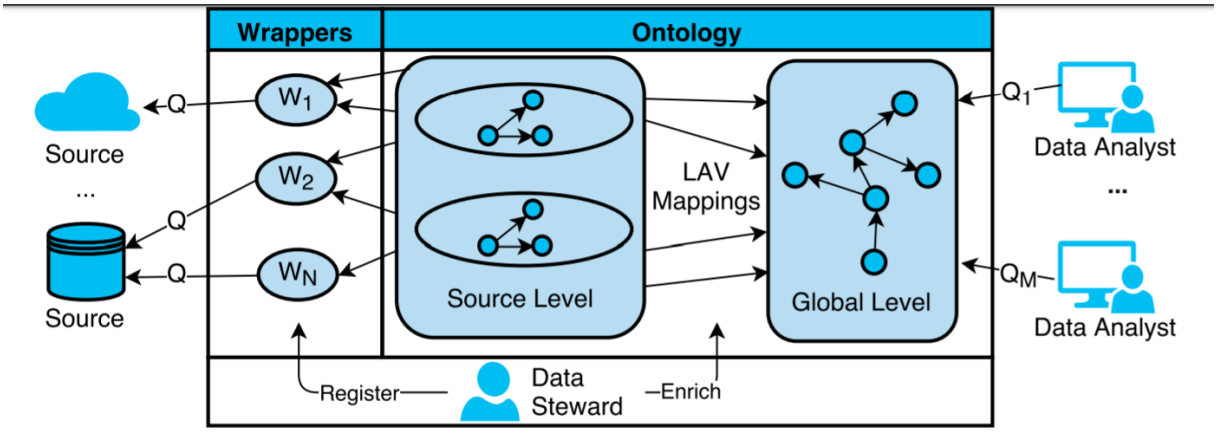


Ilustración 8 - Intervención manual de configuración de envoltorios [31]

En [30] se recapacita precisamente en los problemas que hemos planteado con anterioridad acerca del descontrol y desgobierno que puede derivarse de los ecosistemas *Big Data* si no hacemos uso de ontologías y semánticas apropiadas y un gobierno de datos adecuado, pero donde se necesita de intervención humana continua para el emparejamiento y el enriquecimiento de la información.

2.2.2 Oferta comercial

Estas aproximaciones son aproximaciones estáticas y esa rigidez en la solución, hacen que sean alternativas poco adaptativas y que ante cualquier variación sea necesario un reajuste completo de la implementación de las asociaciones hechas.

Así por ejemplo vemos en *Ontology-Based Data Access (*OBDA) [32], que se sigue el mismo planteamiento estático haciendo uso de una ontología de apoyo (que no tiene por qué ser un estándar de facto, sino que puede ser creada por el usuario), una serie de fuentes de datos y un conjunto de mapeos que son afirmaciones que expresan las relaciones entre la ontología y los datos reales. Es decir, siempre usando un **mismo planteamiento de asociaciones manuales:**

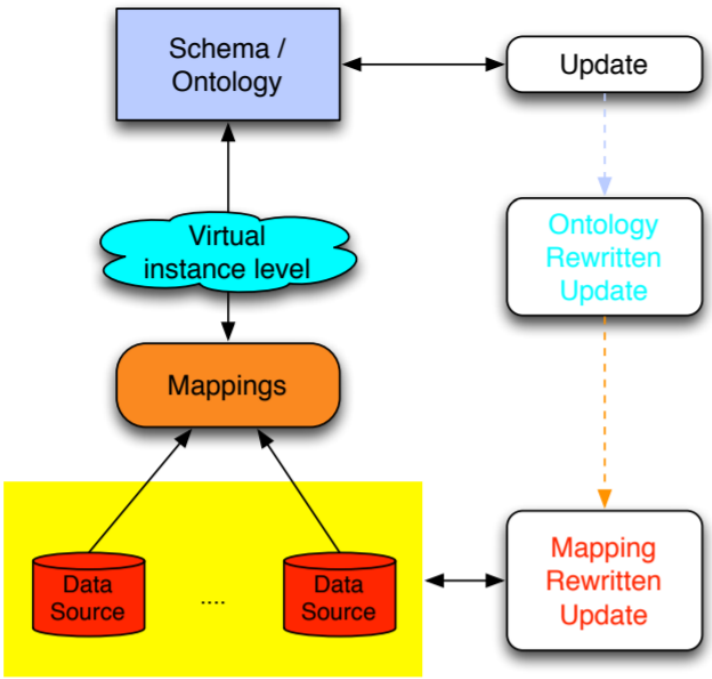


Ilustración 9 - OBDA Arquitectura de alto nivel [32]

OBDA es un sistema originado de la investigación y convertido en un producto comercial denominado *Mastro* [33] y cuya arquitectura lógica propuesta de tres capas:

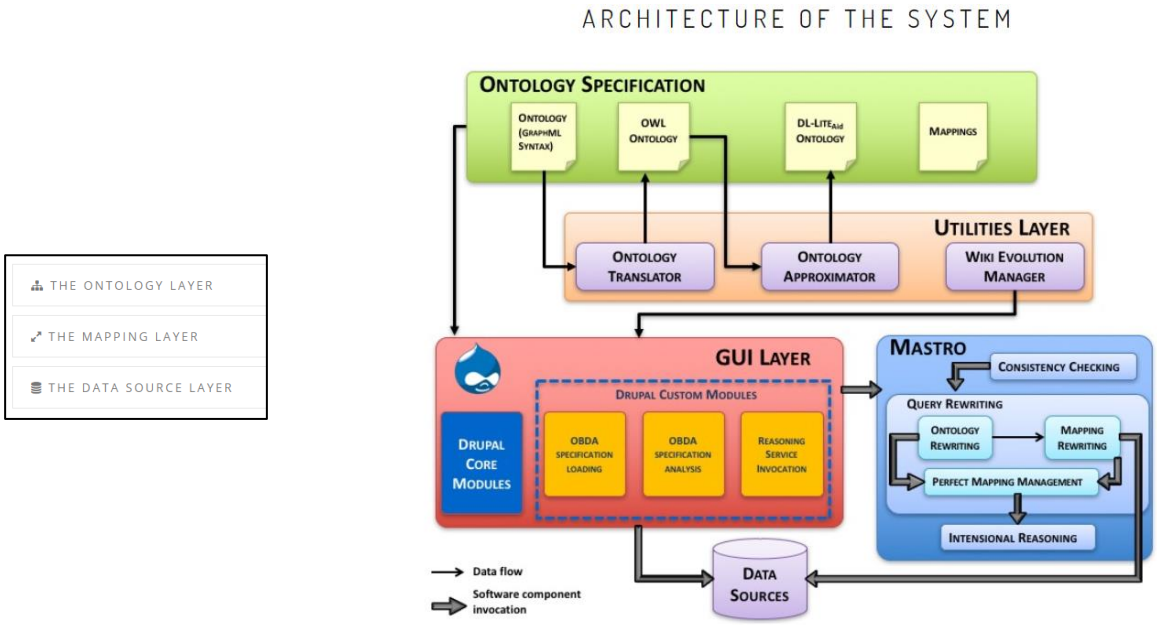


Ilustración 10 – OBDA – Arquitectura de Mastro [33]

se asemeja bastante al planteamiento que queremos abordar en nuestra investigación, sin embargo, nuestro modelo sería complementario como veremos en los siguientes apartados.

2.2.3 Disciplinas involucradas y retos

En todas las investigaciones actuales que se están haciendo en este campo, se observa que siempre se cruzan varias áreas de conocimiento donde finalmente el poder correlacionar texto y significado usando ontologías es el reto que batir. Como se cita en [34] no es una tarea trivial ya que se encuentran varios retos; es difícil capturar la semántica entre la correlación de características, la ausencia de estudios científicos de como correlacionar e integrar características dentro del mundo de *Interpretable Machine Learning (IML) [35]* para generar esas explicaciones semánticas de una manera fácil de entender y, por último, la búsqueda de explicación semántica no suele contar con grandes cantidades de datos y suele ser un proceso complicado. Por ello en [34] se propone la variante *Ontology-based IML* (*OnML*) para generar contenido semántico integrando conocimientos específicos de dominio codificado en ontologías y una extracción de información usando técnicas de *IML*.

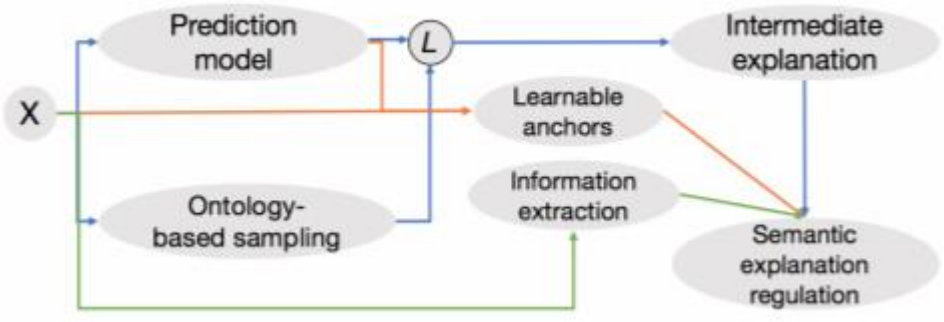


Ilustración 11 - Flujo de la aproximación OnML extraído de [34]

En este sentido en [34] se aborda un posible paso posterior al que queremos abordar en este trabajo a un mayor nivel de profundidad, ya que, si conseguimos asociar el dato técnico a una ontología usando técnicas de *machine [36]* y *deep learning [37]*, la interconexión entre ontologías para extraer significado usando *OnML* puede ser un nuevo campo de investigación para completar el estudio.

Como podemos intuir, en este tratamiento de información para conseguir ese gobierno deseado, siempre tenemos que pensar en el conocimiento necesario para unir significados y así poder asociar los datos a ese significado, así como vemos en [38] donde se explica claramente la misma idea que hemos expuesto aquí, donde los conceptos pueden definirse a priori, de modo que los nuevos datos tienen que ajustarse a los conceptos existentes, o los conceptos pueden ampliarse paso a paso sobre la base de nuevos datos.

Para representar el significado semántico de las fuentes de datos, es necesario establecer un mapeo entre ontología y datos. Como hemos dicho anteriormente, este proceso generalmente se basa en datos al observar los valores reales en los mismos o en etiquetas, según identificadores de datos como nombres de atributos. Sin embargo, en última instancia, **siempre se requiere la intervención humana** para definir si los conceptos reconocidos por los algoritmos son realmente significativos.

Detrás de este paso de asignación de datos a ontologías, siempre hay un cierto esfuerzo, que debe mantenerse lo más bajo posible para el proveedor de datos, y que así continúe su trabajo de la manera más continuada posible y se pueda proporcionar muchas fuentes de datos para el crecimiento del conocimiento.

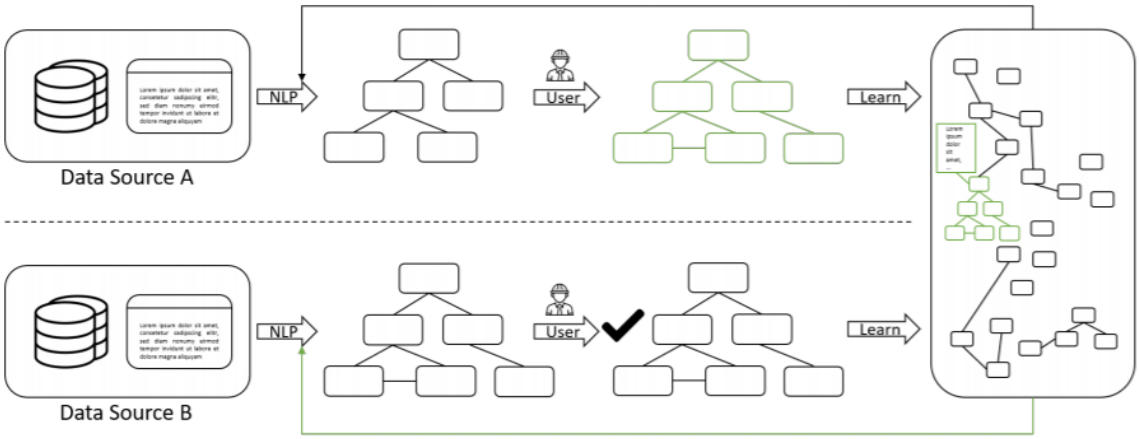


Ilustración 12- Extensión del conocimiento [35]

2.2.4 Mapeo entre Datos, Ontologías, supervisión humana y Procesamiento del lenguaje natural

En 2011, Lenzerini [39] define el paradigma de "O*ntology-Based Data Management (OBDM)*" y sus correspondientes partes teniendo como meta el acceso unificado tanto de los datos como el gobierno de los procesos para manejarlos e integrarlos. Así, OBDM se desglosa en:

- *Ontology-based data access and integration (OBDA)*
- *Ontology-based privacy-aware data access (OBDP)*
- *Ontology-based data quality (OBDQ)*
- *Ontology-based data and service governance (OBDG)*
- *Ontology-based data restructuring (OBDR)*
- *Ontology-based data update (OBDU)*
- *Ontology-based service management (OBDS)*
- *Ontology-based data coordination (OBDC)*

Podemos ver claramente la relación con *DMBoK* [6] siendo este último todavía más rico a nivel de completitud en el campo de la gestión de los datos pero carente del uso en el tratamiento de ontologías.

De manera equivalente a OBDA [32], se fija la arquitectura en la misma aproximación que estamos intentando llevar a cabo en este trabajo, esto es con tres capas bien diferenciadas como hemos visto anteriormente (capa de ontología, capa de orígenes de datos y capa de emparejamiento).

En [38] se especifica la fase temprana de la investigación que se está llevando a cabo al respecto de este campo y añaden una línea complementaria con la idea de usar procesamiento del lenguaje natural (*Natural Language Processing, NLP)* para dotar de otra dimensión a este campo, yendo más allá del valor y las etiquetas de los datos y enriquecerlo con métodos para el aprendizaje de ontologías desde el propio texto [40].

En definitiva, este campo de investigación está en fase muy temprana de obtener resultados en donde se vea realmente el éxito de estas nuevas técnicas de gestión de los datos usando *machine learning*, mapeo de ontologías y en última instancia, enriqueciéndolo haciendo uso de lenguaje natural para así, no digamos metadatar, sino de dotar de significado completo de los textos de un ecosistema empresarial real para el gobierno de los datos.

3 Objetivos y metodología de trabajo

El principal objetivo de nuestro TFM, es demostrar que se puede construir un catálogo de términos de negocio a partir de datos técnicos persistidos en diferentes fuentes, y crear la vinculación de naturaleza técnica con su significado semántico de una manera automatizada mediante técnicas aprendidas de inteligencia artificial para enriquecer un proceso de gobierno de datos corporativo.

3.1 3.1. Objetivo general

Se pretende conseguir una asociación de datos técnicos que persistan en diferentes fuentes de datos de una organización, con conceptos de negocio vinculados a una ontología, usando una inteligencia artificial, construyendo de esta manera una de las piezas fundamentales para un proceso de gobierno de datos como es el repositorio de conceptos de negocio.

Plantearemos el uso de técnicas de *Machine y Deep Learning* para eliminar y evitar soluciones estáticas y manuales como las que hemos visto en el apartado del estado del arte, siendo sustituidas estas tareas, por un sistema artificial modelado que prescinda de las tareas del administrador de datos e ingeniero de configuración de mapeos de la ecuación, y aprovechándonos de los estándares mundiales públicos como las ontologías que hemos mencionado para reglamentar cualquier nuevo dato.

Se desean alcanzar los siguientes objetivos:

- Realizar un prototipo que catalogue datos técnicos con sus correspondientes conceptos de negocio acorde a *FIBO*.
- Generar una simulación de glosario de términos para nuevas entradas que reciba el sistema.
- Introducir, de ser satisfactorio los resultados del prototipo, la solución dentro del proceso del gobierno de datos en una organización con el objetivo de mejorar la productividad y tiempos de despliegue de nuevos términos.

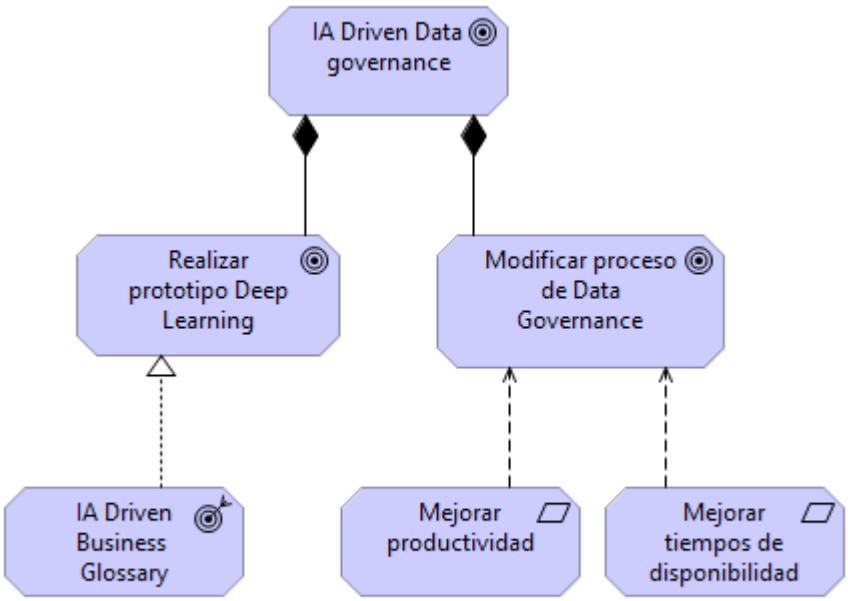


Ilustración 13 - TFM Goal Realization View

3.2 Objetivos específicos

Para lograr los objetivos generales del TFM definimos una serie de objetivos específicos que debemos desarrollar para llegar a un objetivo real:

- Analizar la ontología del dominio elegido. Dominio Bancario. Ontología *FIBO*.
- Descubrir conjunto de datos públicos que nos sirvan de fuentes ficticia para poder desarrollar las pruebas que queremos de clasificación.
- Explorar los datos para poder hacer una preparación de los mismos y tener claro la semántica que representan.
- Asociar, a los conjuntos de datos conseguidos, su correspondiente clase de la ontología que lo representa.
- Diseñar un conjunto de datos de entrenamiento consistente a nivel de significado.

- Plantear qué tipo de técnica de inteligencia artificial es la que mejor se adecua al problema a resolver. En primera instancia pensamos que el problema se resolverá mediante una red neuronal artificial.
- Realizar los experimentos necesarios donde se plantearán diferentes arquitecturas de red, así como inferencia de los mejores híper parámetros para conseguir que la red actué como deseamos.
- Comparar resultados para poder conseguir la mejor red para el objetivo comentado.
- Desarrollar las conclusiones y aplicabilidad real del prototipo en un entorno empresarial.

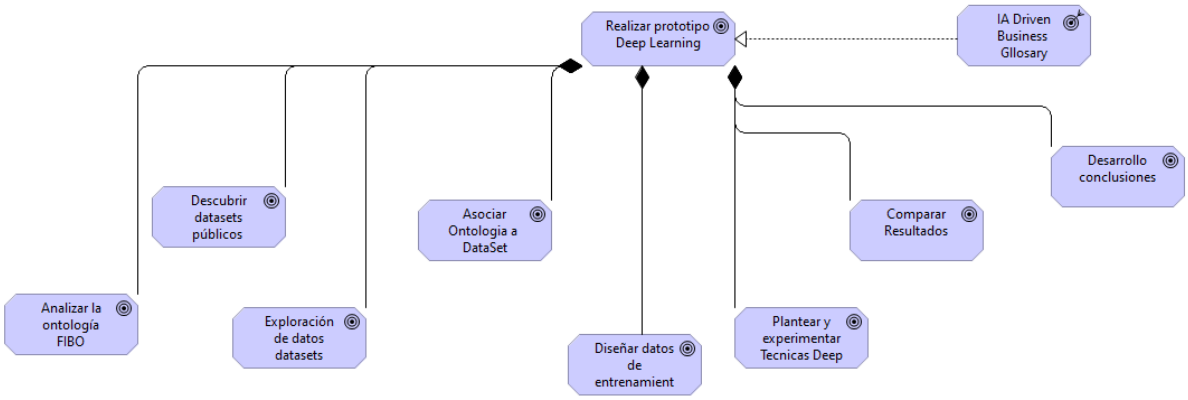


Ilustración 14 - Metas específicas

3.3 Metodología del trabajo

La aplicabilidad del correcto resultado de las técnicas que nuestro TFM pretende resolver, podrían originar un cambio en el uso del glosario de términos corporativos dentro del proceso de gobierno de datos que exista en las organizaciones, puesto que en el propio proceso de validación de la publicación de un dato en el glosario de términos corporativo, este proceso se verá afectado siendo un modelo desplegado el que esta vez tomará la decisión (en base a un umbral de calidad que determinemos) de si un determinado dato está vinculado, o no, con un concepto de la ontología usada.

Un ejemplo de proceso completo del dato. dentro de la compañía, y haciendo foco en el paso de la creación del glosario de términos, puede ser el que sigue:

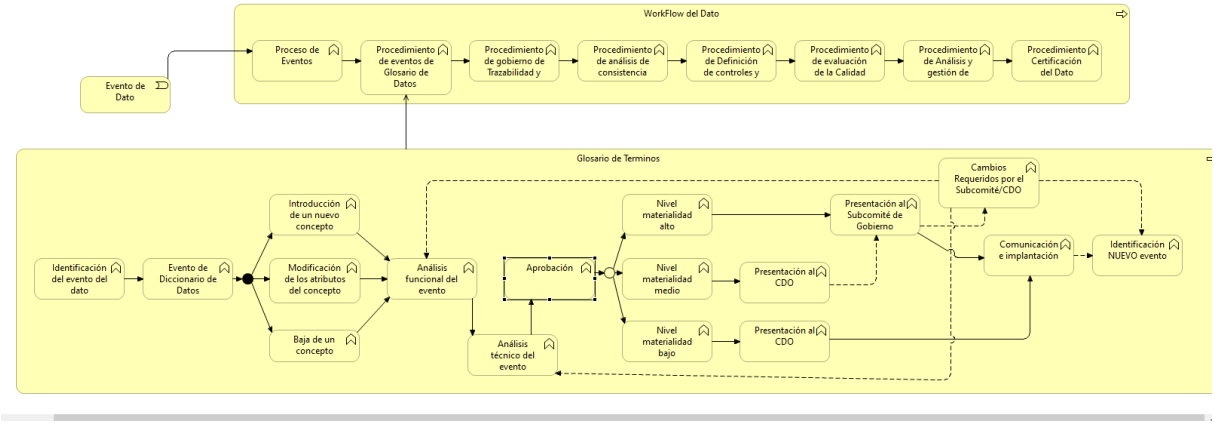


Ilustración 15 - Metodología para un Glosario de Términos corporativo

Vemos que es un proceso complejo y que suele llevar días o semanas, en la mera aprobación para que un nuevo dato y su semántica asociada formen parte del catálogo de conceptos de negocio.

Con nuestro enfoque, esta metodología asociada se alterará/aligerará al introducir este nuevo planteamiento de uso de inteligencia artificial en el proceso de gobierno de datos, concretamente en la gestión de los procedimientos asociados al diccionario de conceptos de negocio.

Nuestra aproximación será una aproximación “de abajo a arriba” (de manera equivalente se podría hacer una aproximación “de arriba a abajo”, es decir desde la definición del concepto de negocio hacia la vinculación técnica de la fuente):

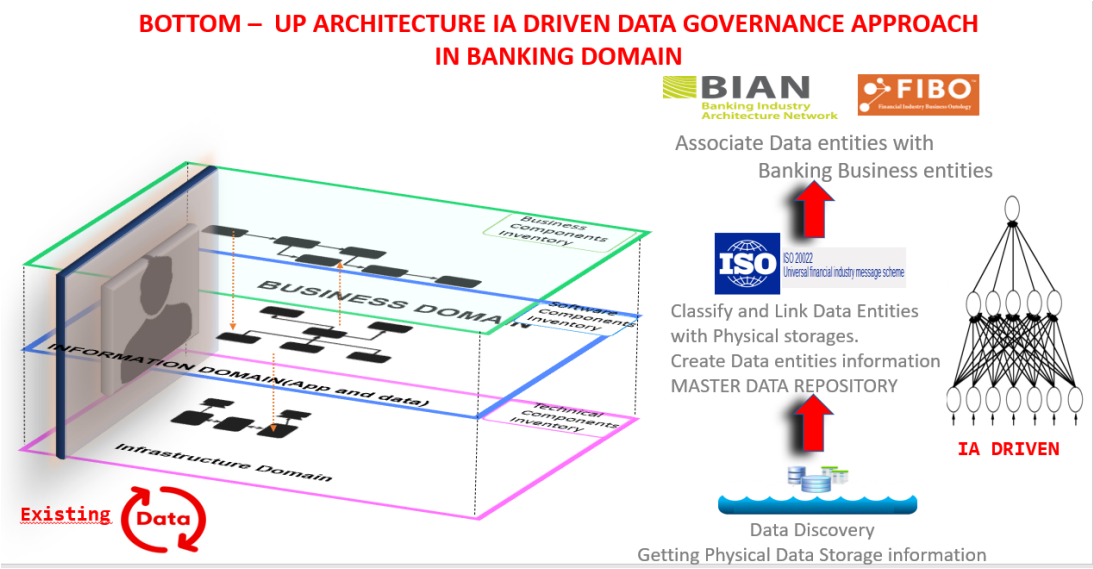


Ilustración 16 - Aproximación "de abajo a arriba" para catalogación dirigida por una IA

Debido a la necesidad de acotar el alcance de este TFM, se vinculará directamente los datos técnicos con las clases de la ontología.

Se pueden encontrar 'puentes' entre diferentes estándares como pueden ser la *ISO 20022* [41] (estándar de mensajes bancarios) con *FIBO* [13] y a su vez con un marco de arquitectura funcional como *Banking Industry Architecture Network* (*BIAN)* [42, 43]. Pero en este TFM lo restringiremos a:

DATO TÉCNICO → CLASE DE LA ONTOLOGÍA DE DOMINIO

Una vez implantado satisfactoriamente un modelo que vincule y asocie los datos técnicos con su correcta clase de la ontología del dominio específico, la metodología interna se vería afectada de la siguiente manera:

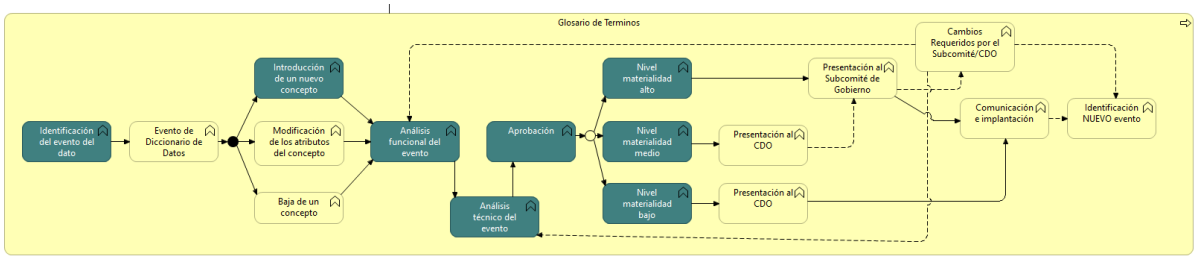


Ilustración 17 - Metodología 'relajada' por uso de IA

Las funciones de negocio de color verde, se verían afectadas por la incorporación de un modelo con un grado de precisión y calidad aprobado por la organización para el cometido que nos ocupa.

Sólo si se persigue por normativa interna, la presentación y aprobación presencial de los nuevos términos descubiertos (para que formen parte del glosario de términos de la corporación), se necesitaría intervención humana. Si la organización lo aprobase, una vez el modelo categorice y asocie (con un umbral de fiabilidad concreto) un término, podría directamente incluir dicho término para ser consumido y consultado de manera automática por toda la organización, eliminando o aligerando enormemente toda la metodología rudimentaria existente.

4 Identificación de requisitos

4.1 Localización del problema en la organización:

En una arquitectura de datos en las compañías, se suele tener un mapa de arquitectura tipo, dónde el lago corporativo suele estar compuesto de varios estadios cada uno con su misión. Generalmente un lago se divide de las siguientes capas lógicas “Ingesta”, “Aterrizaje (*Landing*)”, “Almacenamiento (*Storage*)”, “Escena (*Staging*)” y “Servicio (*Service*)”.

En [30] se expone claramente el problema al que nos enfrentamos a nivel de silos de información y desgobierno completo de los datos al respecto de su significado. La idea “productiva” de este trabajo se situaría justo después de la fase de “Ingesta” y entre la fase de “Aterrizaje” y “Almacenamiento”.

Justo en ese punto, cuando se procesan los datos y se almacenan, ya que seguimos una aproximación de arquitectura ‘de abajo a arriba’, es ahí cuando dispararíamos el consumo de nuestro modelo para que ese activo corporativo, el dato, se catalogue dentro del diccionario y, el gobierno de la organización, lo asocie a una naturaleza semántica estandarizada.

Esos datos, proveniente de las diferentes fuentes de ingesta, ya sea real time o no, es el que queremos usar como input a nuestro modelo. En la estandarización de *Open Business Data lake (O-BDL)* [44], se define de la siguiente manera:

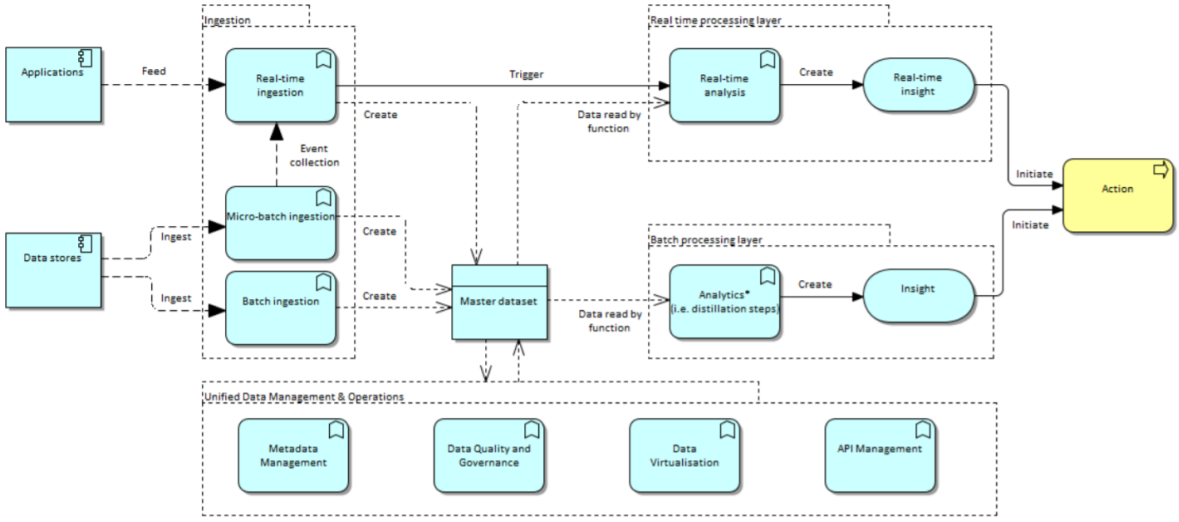


Ilustración 18- O-BDL extraído de [41]

Y donde vemos que es la pieza central de todos los procesos de adquisición y exposición posterior de los datos.

4.2 Requerimientos:

En nuestro ejercicio de investigación, tendremos que usar conjuntos de datos bancarios para simular nuestra entrada al modelo. Dado la naturaleza del problema restringiendo el dominio a un entorno bancario real, y necesitando datos que retraten el mundo bancario, el conseguir un conjunto de datos apropiados, ha sido una tarea bastante difícil ya que son datos que no se disponen al público de una manera gratuita o libre.

Aun así, hemos conseguido un conjunto de datos de [45] vinculado a un banco de la Republica Checa, donde sus datos están anonimizados y se dispone a libre disposición para usos no comerciales.

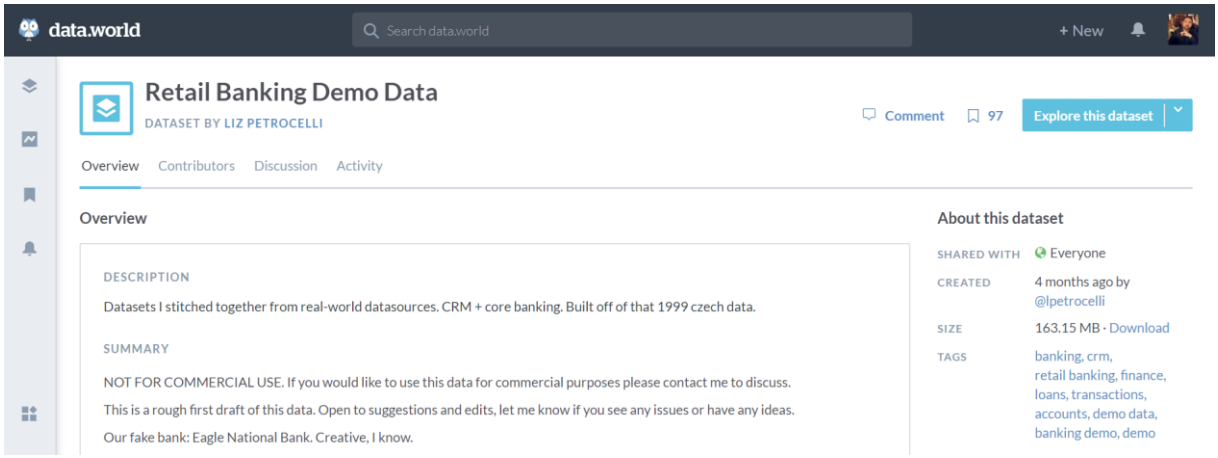
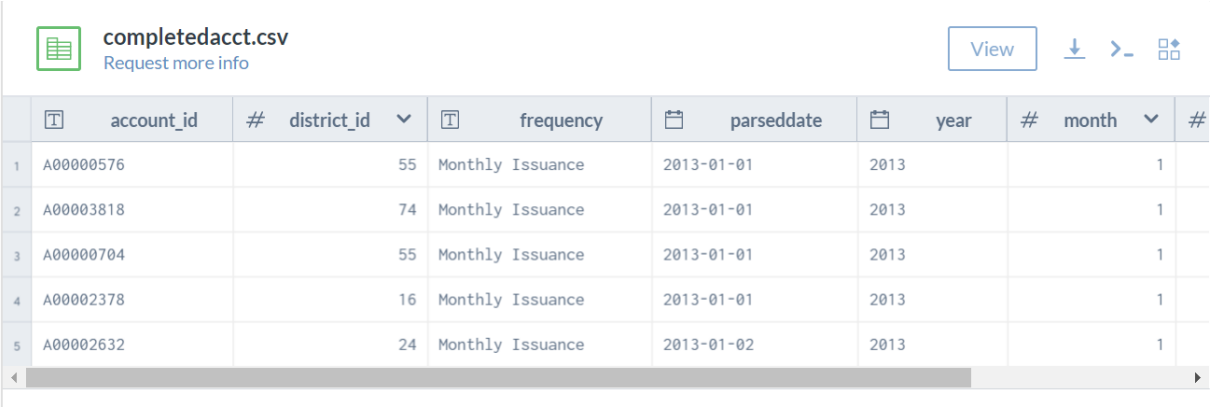


Ilustración 19 - Conjunto de Datasets para simulación

A este conjunto de datos tendremos que añadir, la categoría de concepto de negocio en la que se encuentra cada registro, ya que no se obtiene del mismo.

Esto ha conllevado un trabajo manual para vincular cada conjunto a una clase correctamente identificada en *FIBO* [13], por lo que debemos encontrar la correspondencia adecuada para poder entrenar el modelo que queremos que sea nuestro clasificador de tiempo real de datos. Así por ejemplo para los datos del conjunto de “Cuentas personales” (“*Accounts*”):



completedacct.csv
Request more info
View

	account_id	district_id	frequency	parseddate	year	month
1	A00000576	55	Monthly Issuance	2013-01-01	2013	1
2	A00003818	74	Monthly Issuance	2013-01-01	2013	1
3	A00000704	55	Monthly Issuance	2013-01-01	2013	1
4	A00002378	16	Monthly Issuance	2013-01-01	2013	1
5	A00002632	24	Monthly Issuance	2013-01-02	2013	1

Ilustración 20 - Ejemplo de dataset bancario

tendremos que añadir una columna que identifique su vinculación dentro de la ontología concreta que estemos usando.

Antes de seguir y completar este paso del proceso, es un requerimiento necesario entender la ontología propuesta y cómo funciona *FIBO*.

4.3 Entendiendo FIBO

La Ontología Empresarial de la Industria Financiera *(Financial Industry Business Ontology*) *FIBO[13]*, define los conjuntos de elementos que son de interés en las aplicaciones de negocios financieros y las formas en que esos elementos pueden relacionarse entre sí.

De esta manera, FIBO puede dar significado a cualquier dato (por ejemplo, hojas de cálculo, bases de datos relacionales, documentos XML) que describan el negocio de las finanzas. Es por ello, por lo que este ejercicio es un primer paso que puede brindar la oportunidad para poder desarrollar todo un ecosistema ligado al gobierno de los datos financieros asociándoles el significado correcto y estandarizado a nivel mundial por esta ontología.

Como indican en su web, desde enero de 2020, *EDMC FIBO* [13] ha sido desarrollado por un proceso comunitario abierto con la misión de desarrollar, mantener y promover **estándares de datos globales**, **independientes de plataforma, legibles por máquina y sin ambigüedades que permitan comprender la terminología financiera**, **la federación y la agregación entre sistemas** de datos para mejorar la efectividad de las decisiones, mejorar la eficiencia en la presentación de informes reglamentarios **y acelerar la adopción de capacidades analíticas avanzadas para los servicios financieros.**

FIBO es un conjunto de ontologías. Está organizado en una estructura de directorio jerárquico para organizar las ontologías. Los directorios de nivel superior se denominan

dominios; debajo de eso puede haber uno o dos niveles de subdominio y luego módulos y docenas de ontologías en el nivel inferior:

- Dominio FIBO
    - (Subdominio FIBO)
        - Módulo FIBO
            - Ontología FIBO

Existe una lista de once dominios FIBO empezando con las “Entidades” (*Business Entities*) y terminando con los conceptos asociados financieros relacionados con Seguridad (*Securities*):

Ilustración 21 - Ontologías FIBO

El código de colores que vemos en la imagen, indica el grado de madurez de la ontología, así cada ontología es verde o amarilla. El color verde indica que una ontología tiene un nivel de madurez de "*release*" (liberada), mientras que el amarillo significa que es provisional o informativo. Los dominios o módulos son verdes (amarillos) si contienen solo ontologías verdes (amarillas). Los dominios o módulos son verde-amarillo si incluyen ontologías verdes y amarillas:

- ***Release***: ontologías que se consideran estables y maduras desde una perspectiva de desarrollo.
- **Provisional**: ontologías que se consideran en desarrollo.
- **Informativas**: ontologías que se consideran obsoletas, pero se incluyen con fines informativos porque están referenciadas por algún concepto provisional.

4.4 Asociación de entidad FIBO al conjunto de datos de entrada

Una vez hemos referenciado brevemente la estructura de la ontología a usar, en el tratamiento de los datos buscaremos a qué entidad de negocio ("*Business Entity*") se asocia cada una de las tuplas de nuestros datos de prueba. Como estábamos viendo anteriormente, el dataset "*Accounts*" tendrá una entidad vinculada. Buscamos el término en el buscador que FIBO proporciona y vemos que "***Accounts***" está vinculado con *https://spec.edmcouncil.org/fibo/ontology/FBC/ProductsAndServices/ClientsAndAccounts/Account* y su nombre cualificado (Qname) es ***fibo-fbc-pas-caa:Account***. Este sería el *label* de los elementos del dataset, o valor a predecir que podríamos completar con toda la información que queramos.

Así, por ejemplo, FIBO nos ofrece la siguiente descripción para el término "**Account**":

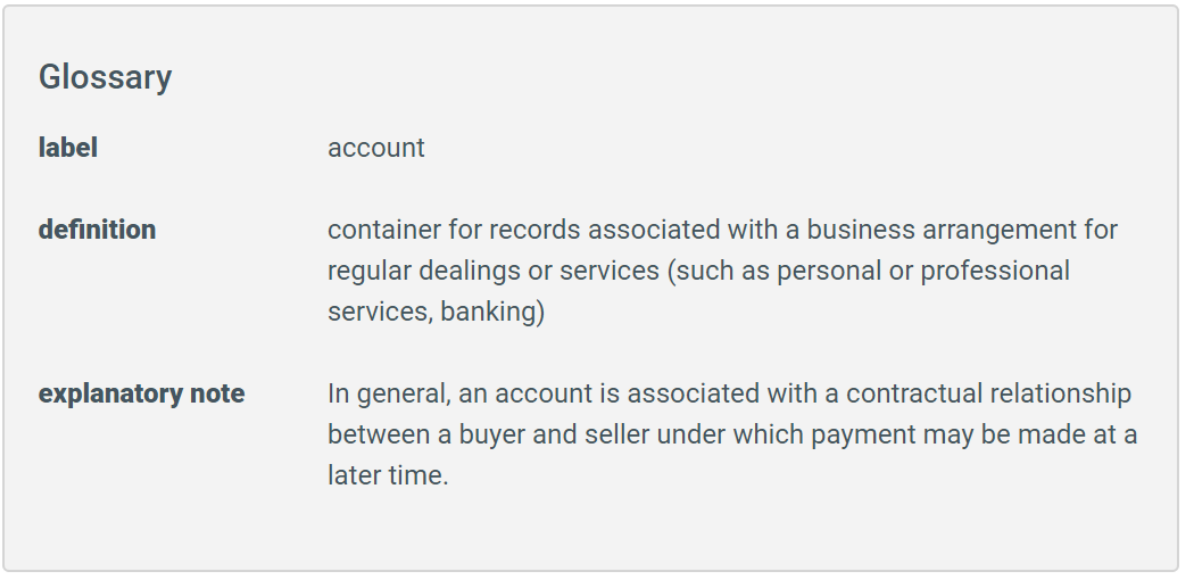


Ilustración 22-FIBO Descripción de Account

Y toda la información relacionada con el término en la ontología dotándole de un significado:

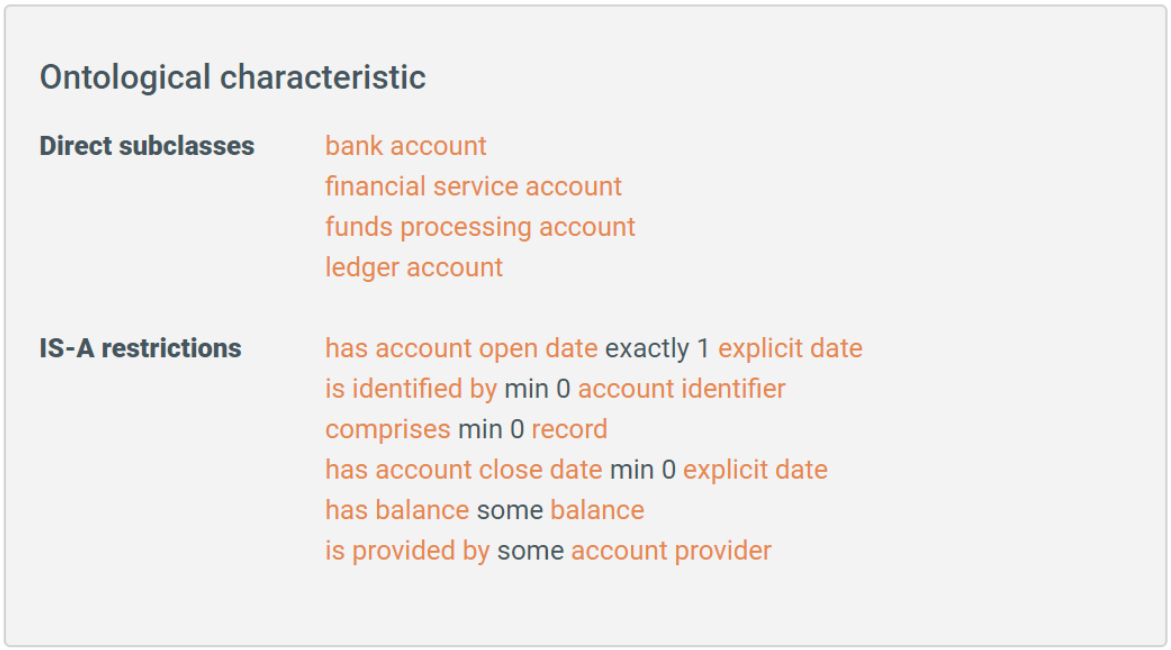


Ilustración 23 - FIBO Caracteristicas Ontológicas del término Account

Inclusive su mapa de relaciones para poder navegar entre los términos vinculados a tal concepto:

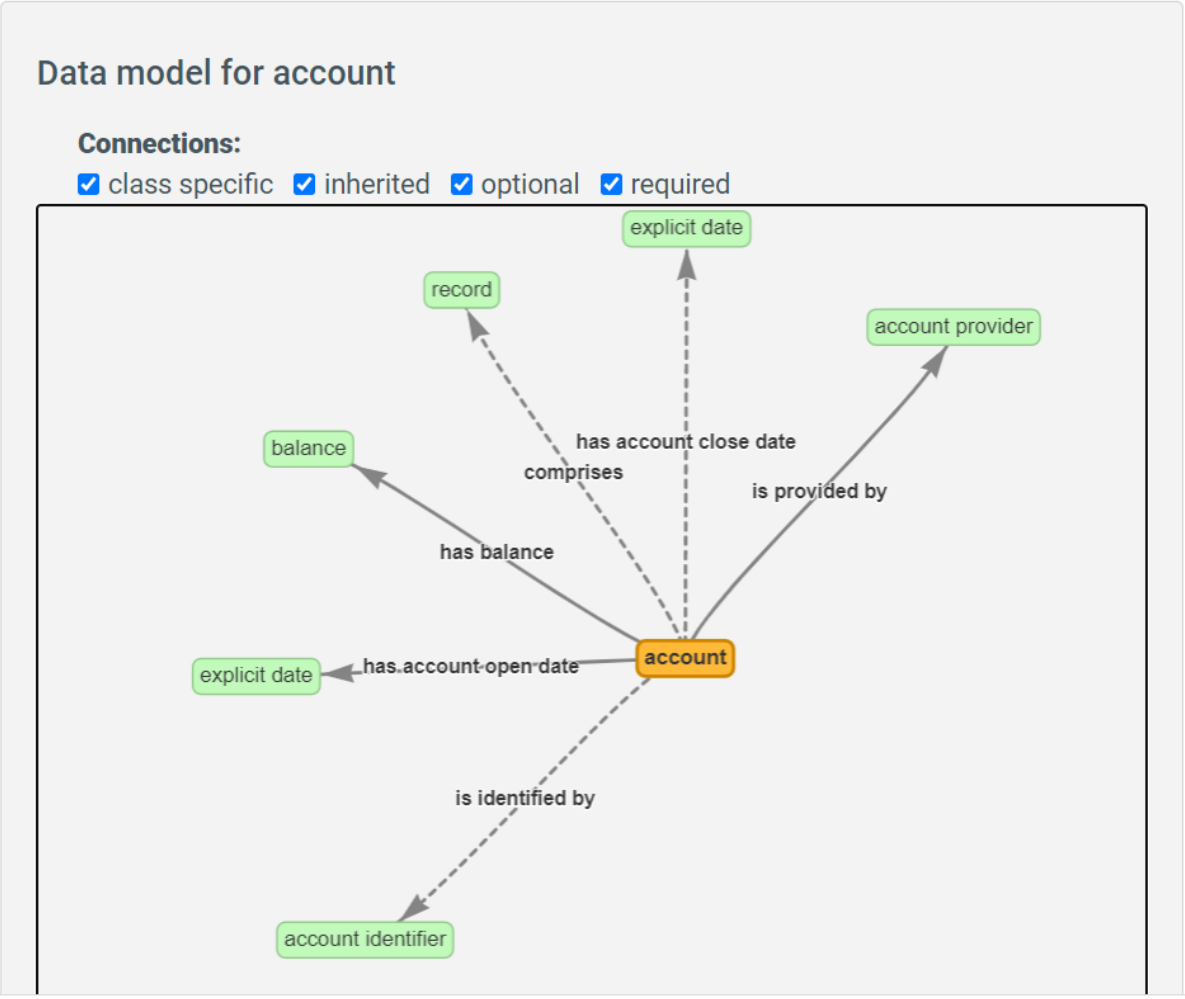


Ilustración 24 FIBO Visualización de la Business Entity Account

FIBO ofrece este visualizador de ontologías, pero igualmente podríamos utilizar herramientas como *Protegé* [46]; herramienta libre de la Universidad de Standford ampliamente conocida y usada en diferentes ámbitos.

Visto este proceso previo, para poder desarrollar nuestra idea, tendremos los siguientes conjuntos de datos vinculados y adaptados con su correspondiente entidad ("*Business Entity*") de FIBO:

Nombre de Dataset	FIBO	Qname
Accounts.csv 4,500 rows, 4 columns	Financial Business and Commerce	fibo-fbc-mod:FBCDomain fibo-fbc-pas-mod:FBCProductsAndServicesModule fibo-fbc-pas-caa:

	FIBO FBC Products and Services Module Clients and Accounts Ontology Account	fibo-fbc-pas-caa:Account
Card.csv 892 rows, 4 columns	Loans Loan Types Module CreditProducts credit card facility	fibo-loan-mod:LOANDomain fibo-loan-typ-mod:LoanTypesModule fibo-loan-typ-cr: fibo-loan-typ-cr:CreditCard
Client.csv 5,369 rows, 3 columns	Foundations Products and Services Products and Services Ontology Client	fibo-fnd-mod:FNDDomain fibo-fnd-pas-mod:ProductsAndServicesModule fibo-fnd-pas-pas: fibo-fnd-pas-pas:Client
Disp.csv 5,369 rows, 4 columns	Loans Loan Contracts Module Loan HMDA Ontology HMDA disposition	fibo-loan-mod:LOANDomain fibo-loan-ln-mod:LoanContractsModule fibo-loan-ln-hmda: fibo-loan-ln-hmda:HMDA_Disposition
Loan.csv	Loans	fibo-loan-mod:LOANDomain

682 rows, 7 columns	Loan Types Module CommercialLoans	fibo-loan-typ-mod:LoanTypesModule fibo-loan-typ-com:
Order.csv 6,471 rows, 6 columns	Foundations Products and Services Payments and Schedules Ontology Payment	fibo-fnd-mod:FNDDomain fibo-fnd-pas-mod:ProductsAndServicesModule fibo-fnd-pas-psch: fibo-fnd-pas-psch:Payment
Trans.csv 1,056,320 rows, 17 columns	Foundations Transactions TransactionEvents transaction undertaking	fibo-fnd-mod:FNDDomain fibo-fnd-txn-mod:TransactionsExtModule fibo-fnd-txn-ev: fibo-fnd-txn-ev:TransactionUndertaking

Cabe destacar el nivel de agregación que estamos asumiendo y pensando para el ejercicio. Es decir, en nuestro ejercicio, dado la escasez de datos de los que disponemos, agregamos el registro de un dataset, por ejemplo, de "*Accounts*" a la entidad de FIBO "*Account*", ahora bien, el registro del dataset está compuesto por diferentes elementos/partículas sobre las que también podría aplicarse el mismo ejercicio, de tal manera que la granularidad asociada sería mucho más completa para ese diccionario de términos generado automáticamente.

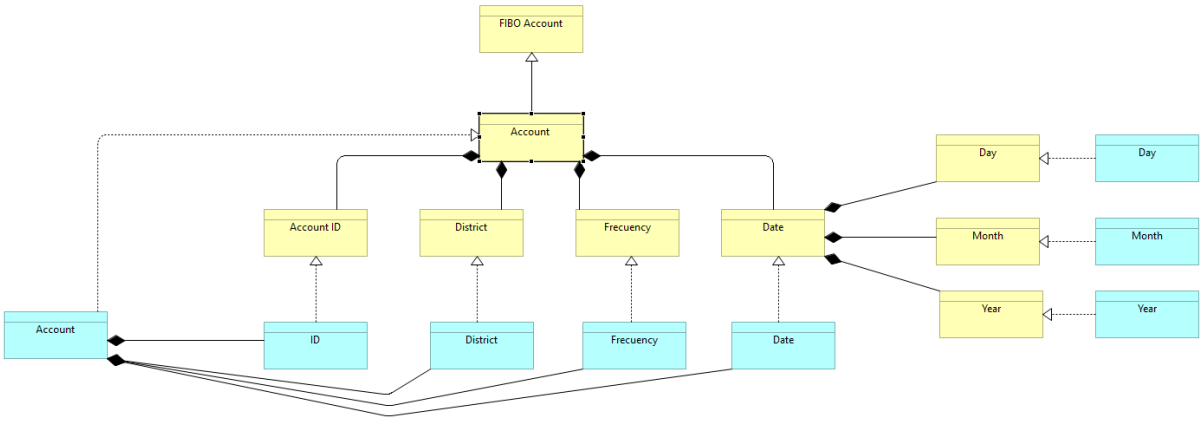


Ilustración 25 - Composición de una Business Entity

Es más, es en este punto en el que profundizando en cada una de las partículas de la composición llegaríamos a la atomicidad del elemento (realmente conseguiríamos una unión de ontologías y relación entre las mismas).

Por ejemplo, el concepto “día” (“*Day*”) no es “propio” de FIBO, pero sí lo referencia uniendo en este punto, por ejemplo, la definición de “*day*” con su código estandarizado de la *Object Management Group (OMG)* de la ISO-639-2 [47]:

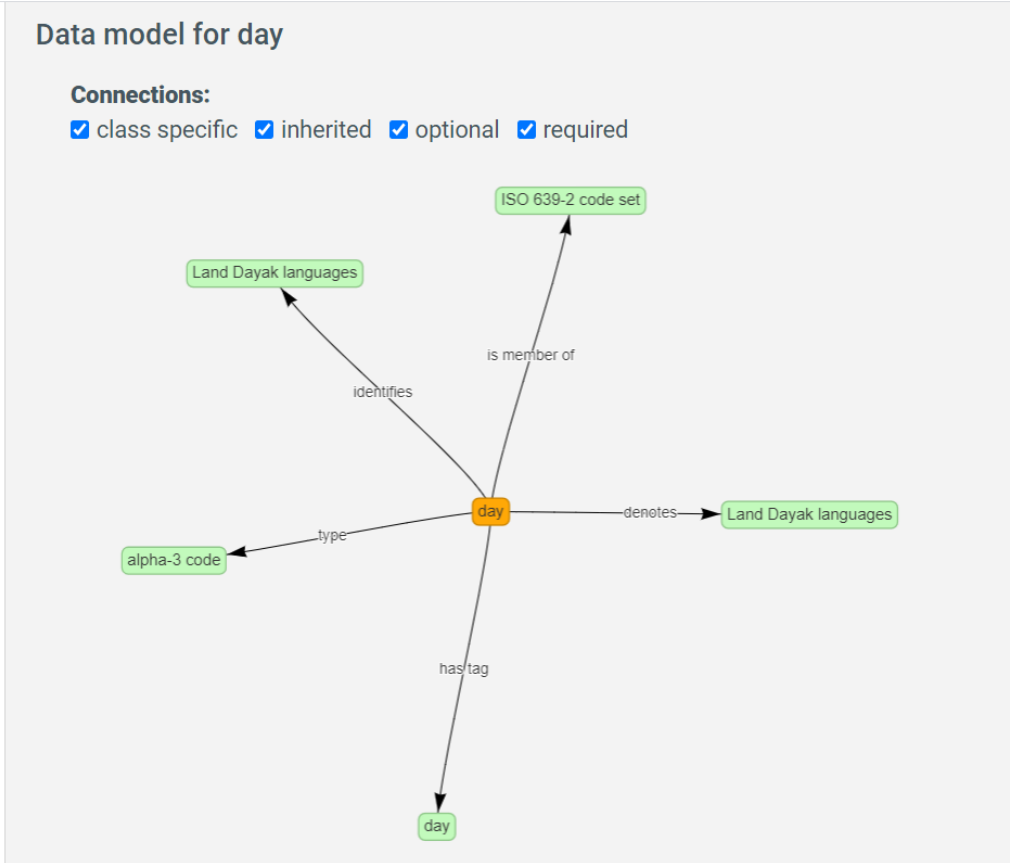


Ilustración 26 Relaciones de la partícula "day"

Correspondiendo a su URI completa de otro dominio estándar https://www.omg.org/spec/LCC/Languages/ISO639-2-LanguageCodes/day, y aquí es donde tendríamos la relación de ontologías estandarizadas y relacionadas con términos de negocio. Si pensamos en esta idea, simple pero poderosa, finalmente tendríamos relacionado todo el conocimiento funcional de cada dominio concreto, con otros dominios en determinados conceptos que serían una especie de **centroides de interconexión entre dominios de conocimiento de ontologías**.

Debido al alcance de este trabajo, debemos restringir las expectativas para poder obtener resultados de asociación entre los datasets agregados que tenemos, con su correspondiente clasificación de FIBO, pero la idea seguiría siendo válida y posible de aplicar a cualquier grado de elemento, ya sea partícula o compuesto de las mismas, para finalmente tener ese mapa de activos de datos correctamente estructurado y formado, y disponible para consumir de manera simple por parte de todos los usuarios de la organización.

5 Descripción de la herramienta software desarrollada

En el proceso de desarrollo de la herramienta creada para simular la viabilidad de la solución identificada en este TFM, hemos seguido una metodología **CRISP-DM** [48] puesto que es la que mejor se adapta a este tipo de desarrollos de investigación individual y está estrechamente vinculado con el propósito de este trabajo.

A nivel de desarrollo software como tal, hemos optado por las utilidades, técnicas, herramientas y aprendizajes obtenidos a lo largo del máster cursado; por ello hemos usado Python y Jupyter Notebook para codificar técnicas de aprendizaje automático, procesamiento de lenguaje natural y sistemas cognitivos artificiales.

5.1 Proceso de desarrollo - metodología

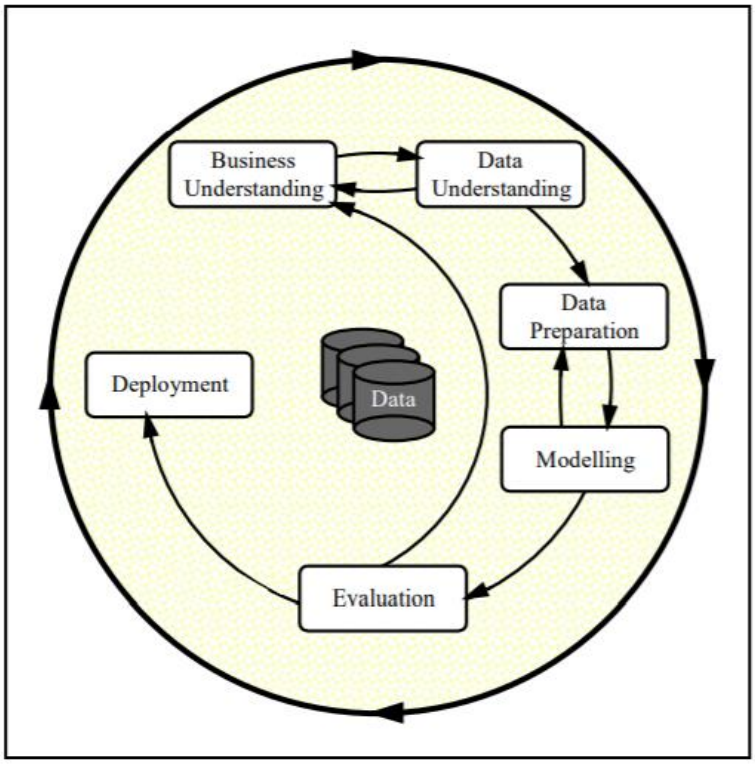


Ilustración 27 - Fases de CRISP-DM [26]

Como cualquier modo de trabajo para este tipo de desarrollos software asociados a inteligencia artificial, y concretamente en este ámbito que estamos tratando, a la inteligencia de datos, procesamiento de lenguaje natural y técnicas de clasificación y aprendizaje

profundo, el entendimiento de los datos y los conceptos de negocio vinculados (fases *Business y Data Understanding en la figura*), el tratamiento y preparación de los mismos (*Data Preparation*) para ir a buscar, o mejor dicho, descubrir los modelos que puedan ajustarse o dar unos buenos resultados (*Modelling*) es la forma con la que hemos trabajado y la que hemos visto que encaja mejor en nuestra propuesta.

La última fase de CRISP-DM, el despliegue ("D*eployment"*) de la solución, como es lógico, radicará en ejecuciones locales para mostrar los resultados obtenidos en este estudio.

Así mismo, debemos pensar en otro tipo de despliegue en la organización, que sería el de vincular el del software en el proceso metodológico del gobierno de datos y que veremos más adelante.

En este apartado ahondaremos en el software desarrollado puesto que es el objetivo.

5.2 Business Understanding y Data Understanding

En el apartado anterior hemos profundizado en el entendimiento de los datos a nivel de entidades de negocio y su nombre cualificado en una ontología estandarizada como es FIBO.

El entendimiento de estos es muy importante, puesto que debemos saber a qué entidades y conceptos se refiere cada una de las partículas de la composición de los registros de los *dataset*. En este caso, como también indicamos en apartados anteriores, y dado el alcance de este trabajo, hemos decidido vincular los datos de manera agregada al concepto de negocio del *dataset* individual donde reside (la tabla).

Una vez fijado este principio en el desarrollo, sabemos que cualquier dato incluido en su *dataset*, forma parte del lenguaje del concepto de negocio al que representa, y que debería asociarse en la clasificación que queremos conseguir a dicha entidad de la ontología estándar a asociar (en este caso a una ontología bancaria al ser datasets demo de un banco minorista).

5.3 Data Preparation

Una vez adquiridos esos datos, y entendiendo la estrategia que queremos afrontar, siendo estos datos de los *datasets* los que conformarán en su totalidad nuestro vocabulario para la prueba, debemos unificarlo en un único dataset que sea sobre el cual intentaremos efectuar la clasificación de la clase de la ontología concreta como ya adelantábamos anteriormente.

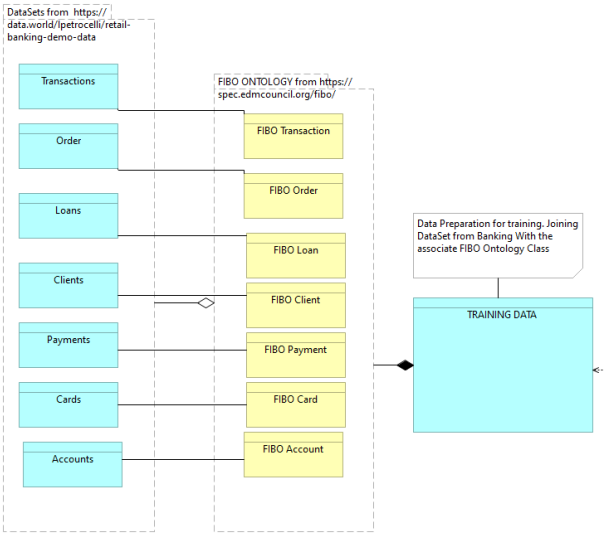


Ilustración 28 Preparación de los datos

```
#-------------------------------------------------------------------------------------
def loadFile(fileName):
    df = pd.read_csv('./datasets/' + fileName, encoding='utf-8')
    return df
#-------------------------------------------------------------------
```

```
#-----------------------------------
def addClass(data, retailClass, Qname):
    #Como estas van a ser las variables a categorizar (predecir) forzamos a que estén en un índice
    # para posteriormente poder efectuar el entrenamiento.
    data['Class'] = retailClass
    return data
#-----------------------------------
```

```
# Datasets con todos los registros que hemos encontrado para poder hacer la prueba de IA datadriven DG --------------
retailDatasets = {'completedacct.csv': ['0','fibo-fbc-pas-caa:Account'], #account
                  'completedcard.csv': ['1', 'fibo-loan-typ-cr:CreditCard'],#credit card facility
                  'completedclient.csv' : ['2','fibo-fnd-pas-pas:Client'] ,#client
                  'completeddisposition.csv' : ['3', 'fibo-loan-ln-hmda:HMDA_Disposition'] ,#HMDA disposition
                  'completedloan.csv' : ['4', 'fibo-loan-typ-com:CommercialLoan'] ,#CommercialLoans
                  'completedorder.csv' : ['5', 'fibo-fnd-pas-psch:Payment'], #orders
                  'completedtrans_lite.csv' : ['6', 'fibo-fnd-txn-ev:TransactionUndertaking' ]#payment
                 }

```

```
COLUMNS = ['account_id','district_id','frequency','parseddate','year','month','day','date',
           'card_id','disp_id','type','fulldate','client_id','sex','age','social','first',
           'middle','last','phone','email','address_1','address_2','city','state','zipcode','district_id',
           'disp_id','type','loan_id','amount','duration','payments','status','location','purpose',
           'order_id','bank_to','account_to','k_symbol','column_a','trans_id','operation','balance','bank',
           'account','fulltime','fulldatewithtime','Class']

```

Ilustración 29 - Código prototipo - Estructura de Datos

Tras recorrer todos los *dataset* de "*Retail bank*", y agregarlos para confeccionar nuestro vocabulario completo, asignamos las clases asociadas a cada registro para poder efectuar el entrenamiento.

```
fileName = 'retailData.csv' # Ficheor donde tendremos los datos finales de las sentencias de retail
finalRetailData = pd.DataFrame(columns=['Sentence','Class']) #estructura fincal- "Frase de registros" y "Clase"

if not(path.exists(fileName)) : # Como es pesado el proceso de gneracion de dataset, comprobamos que exista y si no lo c
    #-------------------------------------------------------------------------------------------------------------------
    # Recorremos todos los datasets fuentes recogidos de https://data.world/lpetrocelli/retail-banking-demo-data
    # y le añadimos las clases de la ontologia que debería aprender por registro asociado
    frames = []
    for data in retailDatasets:
        #Cargamos los datos en un dataframe de Pandas
        df = loadFile(data)
        # Añadimos la clase asociada al registro para el entrenamiento supervisado
        # y Añadimos tambien el nombre cualificado de la ontologia para un posible uso futuro (visualizacion por ejemplo
        addClass(df, retailDatasets[data][0], retailDatasets[data][1])
        print("DATA:" + str(data))
        frames.append(df)

    #Agregado de todos los términos posibles en nuestro diccionario de retail
    retailDataFrame = pd.DataFrame(columns=COLUMNS)
    retailDataFrame = pd.concat(frames, axis=0, ignore_index=True, keys=None,
                      levels=None, names=None, verify_integrity=False, copy=False, sort=False)

    print("Asociando....")
    #Generamos las "frases" con cada uno de los terminos (particulas de los registros)
    retailDataFrame['Sentence'] = retailDataFrame.apply(lambda row: valuation_formula(row), axis=1)
    #Guardamos todos los registros creados conjunto con la sentencia final y la clase
    retailDataFrame.to_csv('allData.csv', index=False, header=True, decimal=',')
    print("Generando.....")
    #Dataset más ligero que sera la fuente para las tecnicas que usaremos
    finalRetailData['Sentence'] = retailDataFrame['Sentence'];
    finalRetailData['Class'] = retailDataFrame['Class'];
    finalRetailData.to_csv(fileName, index=True, header=True)
    print("Generado.")

else: #Si ya lo hemos generado lo cargamos
    finalRetailData =  pd.read_csv(fileName, encoding='utf-8')
```

Ilustración 30 - Preparación de los datos – Generación del Dataset

En este paso, ya tenemos generado nuestro contexto y ámbito de pruebas para poder usar técnicas vistas y así, ver cuán de buena puede ser esta codificación del problema para la resolución del tema que nos ocupa.

5.4 Modeling

Para solucionar la necesidad de catalogación y generación automática de un catálogo de términos embebido dentro del proceso de gobierno de datos de una compañía, como hemos visto, lo hemos “reducido por el alcance” a un problema de clasificación, donde las clases que nuestro modelo debe “aprender”, son entidades de una ontología estándar, en nuestro caso de un entorno bancario, FIBO. Recordemos que esto es totalmente aplicable a otro dominio de conocimiento como hemos visto en apartados anteriores.

Hemos optado por 2 aproximaciones para resolver la necesidad y ver posibles soluciones al problema: clasificador Naive Bayes y aprendizaje profundo.

Antes de crear los modelos, es fundamental entender la aproximación que hemos seguido en el planteamiento: nuestra primera aproximación **es usar los propios registros de los datasets usándolos como si fueran frases textuales**; es decir hemos concatenado los valores de cada una de las columnas del data set, asociándolo a lo que serían nuestros textos

a clasificar en un problema de procesamiento de lenguaje natural y así poder usar las técnicas aprendidas.

```
from sklearn.model_selection import train_test_split

sentences = finalRetailData['Sentence'].values # Obtenermos todas las frases
y = finalRetailData['Class'].values # recogemos las clases para la calsificacion

#Separamos los datos entre entreno y test, dejando un 75% de entreno y un 25% para test
sentences_train, sentences_test, y_train, y_test = train_test_split(sentences, y, test_size=0.25, random_state=1000)
```

```
print(str(len(sentences_train)) + " -- " + str(len(y_train)))
print(str(len(sentences_test)) + " -- " + str(len(y_test)))
```

```
94319 -- 94319
31440 -- 31440
```

```
from sklearn.feature_extraction.text import CountVectorizer

#Tokenizamos y extraemos las caracteristicas
vectorizer = CountVectorizer()
vectorizer.fit(sentences_train)
# para poder generar y entrenar el modelo con las matrices dispersas que genera CountVectorizer()
X_train = vectorizer.transform(sentences_train)
X_test  = vectorizer.transform(sentences_test)
X_train
```

```
<94319x194086 sparse matrix of type '<class 'numpy.int64'>'
        with 2583136 stored elements in Compressed Sparse Row format>
```

Ilustración 31 - Código prototipo - Tokenización de Datos de entreno y test

En este caso hemos optado por una aproximación de bolsas de palabras (*Bag of Words, BOW*) donde tendremos nuestro vocabulario que conformarán las palabras de las frases extraídas del conjunto de datos agregado, resultado de la fase de preparación, y la tokenización de cada una de ellas, resultando en vectores de matrices dispersas que usaremos como datos de entreno y de entrada para la clasificación.

5.4.1 Clasificador Naive Bayes:

Una vez tenemos los datos en vectores tokenizados y en matrices dispersas usando las clases de *CountVectorizer*, pasamos a aplicar un modelo de clasificación sencillo para ver el grado de acierto en la clasificación. Para ello creamos un modelo y aplicamos la predicción:

```
#Vamso a usar a priori un clasificador Naive Bayes como primer paso
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
MultinomialNB()
```

```
5]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
#y Hacemos la prediccion del conjunto de pruebas
y_pred = classifier.predict(X_test)
```

Ilustración 32 - Código prototipo - Clasificación Naive Bayes

Con esta aproximación vemos los siguientes resultados alcanzados usando la matriz de confusión y el nivel de precisión tras la ejecución:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[  757     0     0     0     0     0   350]
 [  116     0     0     0     0     0   113]
 [    0     0  1369     0     0     0     0]
 [    0     0     0  1313     0     0     0]
 [  100     0     0     0     0     0    70]
 [    0     0     0     0     0  1595   104]
 [    0     0     0     0     0     0 25553]]
```

```
#Confirmamos la confianza del modelo usando metricas proporcionadas por las librerias
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9728689567430026
```

Ilustración 33 - Código prototipo - Resultados Naive Bayes

Observamos que obtenemos una confianza del 97% usando esta técnica “sencilla” de clasificación, por lo que la aproximación que hemos llevado a cabo con estos datos tratados de esta forma, es más que sobresaliente para la resolución del problema que hemos acotado.

5.4.2 Clasificación usando aprendizaje profundo

Para comparar con otras técnicas de inteligencia artificial, cambiamos la aproximación de modelos dándole un poco más de generalidad a la hora de poder aprender la clasificación, e introducimos el entrenamiento con una red neuronal (sencilla) que aprenda a clasificar las clases en función de los datos que ya hemos tratado anteriormente.

```
input_dim = X_train.shape[1]  # Numero de características
print(input_dim)
```

```
194086
```

```
y_train = keras.utils.to_categorical(y_train, NUM_CLASSES)
y_test = keras.utils.to_categorical(y_test, NUM_CLASSES)

model = Sequential()
model.add(Dense(10, input_dim=input_dim, activation='relu'))
model.add(Dense(NUM_CLASSES, activation='softmax'))
```

Ilustración 34 - Código Prototipo - Red Neuronal con texto tokenizado

Con esta sencilla topología de red, introduciendo únicamente una capa oculta de 10 neuronas, y un vector de 194086 características tokenizadas (75% de los datos) y con unos valores básicos de parametrización estándar para la pérdida y el optimizador:

```
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 10)                1940870
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 77
=================================================================
Total params: 1,940,947
Trainable params: 1,940,947
Non-trainable params: 0
_________________________________________________________________
```

Ilustración 35 - Código prototipo - Topología Red Neuronal I

Observamos los siguientes resultados para 5 epochs:

```
start = time.time()
history = model.fit(X_train, y_train,
                    epochs=5,
                    verbose=True,
                    validation_split=0.20, # validation_data=(X_test, y_test),
                    batch_size=100)
end = time.time()
print("Execution Time:" + str(end - start) + " seconds")

```

```
Train on 75455 samples, validate on 18864 samples
Epoch 1/5
75455/75455 [==============================] - 251s 3ms/step - loss: 0.0998 - accuracy: 0.9626 - val_loss: 0.0625 - val_accu
racy: 0.9724
Epoch 2/5
75455/75455 [==============================] - 237s 3ms/step - loss: 0.0464 - accuracy: 0.9844 - val_loss: 0.0372 - val_accu
racy: 0.9863
Epoch 3/5
75455/75455 [==============================] - 238s 3ms/step - loss: 0.0308 - accuracy: 0.9868 - val_loss: 0.0277 - val_accu
racy: 0.9864
Epoch 4/5
75455/75455 [==============================] - 243s 3ms/step - loss: 0.0221 - accuracy: 0.9930 - val_loss: 0.0191 - val_accu
racy: 0.9980
Epoch 5/5
75455/75455 [==============================] - 244s 3ms/step - loss: 0.0144 - accuracy: 0.9981 - val_loss: 0.0125 - val_accu
racy: 0.9980
Execution Time:1213.661636352539 seconds
```

```
loss, accuracy = model.evaluate(X_train, y_train, verbose=False)
print("Precisión Entrenamiento: {:.4f}".format(accuracy))
loss, accuracy = model.evaluate(X_test, y_test, verbose=False)
print("Precisión Prueba:  {:.4f}".format(accuracy))
```

```
Precisión Entrenamiento: 0.9981
Precisión Prueba:  0.9979
```

Ilustración 36 - Código Prototipo - Resultados Red Neuronal I

Vemos que la precisión es extrema (sobreajuste u *"overfitting"*) y que el tiempo de ejecución de entrenamiento es considerable (20 minutos para 5 epochs):

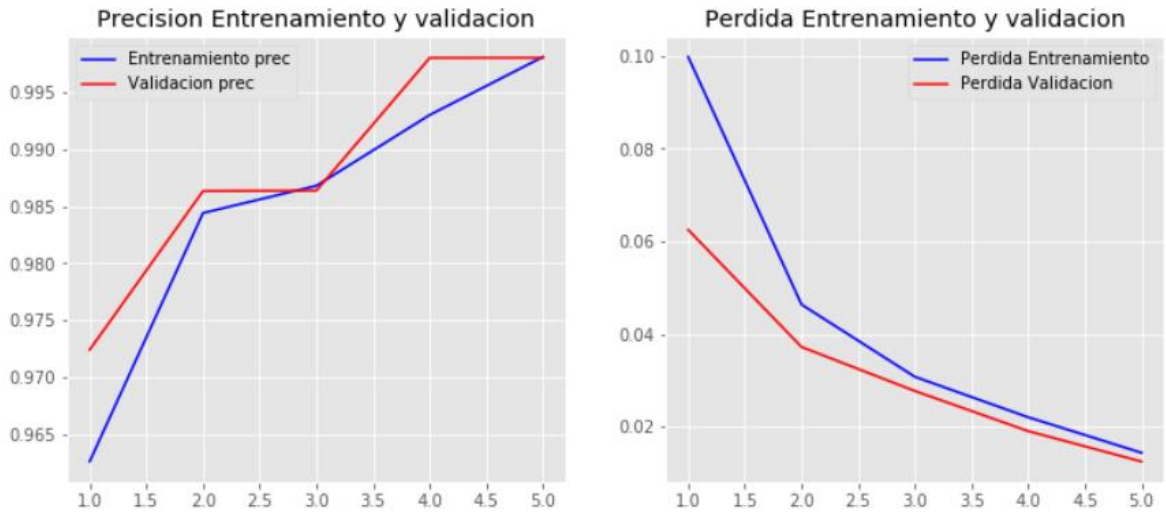


Ilustración 37 - Código Prototipo - Graficas Accuracy y Loss para la Red Neuronal I

5.4.3 Clasificación usando aprendizaje profundo (segunda aproximación)

Dada la naturaleza del entrenamiento y el tratamiento de los datos usando el propio texto de los registros asociados al conjunto de datos y su tokenización como característica, hemos querido comprobar si podemos alcanzar resultados similares optimizando el tiempo de entrenamiento para la red anterior, y simplificando la topología y parámetros de la misma.

Para ello, hemos pensado en una variante del uso de los datos de entrada (recordemos frases de nuestro “mini” vocabulario de un banco minorista) usando la idea de entrenamiento de redes neuronales para la clasificación de imágenes.

La aproximación llevada a cabo ha sido interpretar que una frase corresponde a un “pixel” en técnicas de visión por computador, e intentar transformar esas características a un vector numérico que, de alguna manera, represente a esa entrada, al igual que los valores de nivel de gris en una imagen en blanco y negro, o los valores de rojo, verde y azul (*Red Green blue, RGB)* de un pixel en una imagen a color, pues un valor numérico calculado que nos represente la entrada a la red en cada momento y lógicamente su clase asociada sin alterar.

Para ello, nos apoyamos otra vez en la clase *CounterVectorize()* de *sckitlearn* y fijamos el número de características a un valor concreto (48 en este caso) para usar una función Hash (recordemos que el hash es unidireccional) que transformará y representará la entrada de los registros que queremos clasificar:

```
from sklearn.model_selection import train_test_split

sentences_Hash = retailNumeric['Sentence'].values # Obtenermos todas las frases
y_Hash = retailNumeric['Class'].values # recogemos las clases para la calsificacion

from sklearn.feature_extraction.text import HashingVectorizer
# crear la transformación
vectorizer = HashingVectorizer(n_features=48)
# documento codificado
vector = vectorizer.transform(sentences_Hash)
# resumir vector codificado
print(vector.shape)
print(vector.toarray())

```

```
(125759, 48)
[[ 0.08267216  0.02066804  0.         ...  0.          0.
   0.        ]
 [ 0.08281378  0.          0.         ...  0.          0.
   0.        ]
 [ 0.08267216  0.          0.         ...  0.          0.
   0.02066804]
 ...
 [ 0.         -0.02449245  0.         ...  0.         -0.09796979
   0.        ]
 [ 0.         -0.0249766   0.         ... -0.0249766  -0.09990638
  -0.0249766 ]
 [ 0.          0.04993762 -0.02496881 ... -0.02496881 -0.09987523
   0.        ]]
```

Ilustración 38 - Código Prototipo - HashVector para Red Neuronal II

Y volvemos a obtener los datos acordes a esta idea de transformar texto de entrada para la red:

```
sentencesHash = vector # Obtenermos todas las frases codificadas en un hash
yHash = finalRetailData['Class'].values # recogemos las clases para la calsificacion
#Separamos los datos entre entreno y test, dejando un 75% de entreno y un 25% para test
XHash_train, XHash_test, yHash_train, yHash_test = train_test_split(sentencesHash, yHash, test_size=0.25, random_state=1
```

Ilustración 39 - Código Prototipo - Datos de entreno y Test para Red Neuronal II

Obteniendo una red mucho más “ligera” a nivel de parámetros de entrada, y usando la misma topología de capa oculta y salida, pero con una reducción considerable de los parámetros a optimizar:

```
modelHash.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
modelHash.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 10)                490
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 77
=================================================================
Total params: 567
Trainable params: 567
Non-trainable params: 0
_________________________________________________________________
```

Ilustración 40 - Código prototipo - Topologia Red Neuronal II

Y con unas métricas de resolución de clasificación equiparable a los métodos anteriores testados y con un tiempo de ejecución excelente:

```
start= time.time()
historyHash = modelHash.fit(XHash_train, yHash_train,
                    epochs=10,
                    verbose=True,
                    validation_split=0.20, # validation_data=(X_test, y_test),
                    batch_size=100)
end = time.time()
print("Execution Time:" + str(end - start) + " seconds")
```

```
Execution Time:23.99897289276123 seconds
```

```
loss, accuracy = modelHash.evaluate(XHash_train, yHash_train, verbose=False)
print("Precisión Entrenamiento: {:.4f}".format(accuracy))
loss, accuracy = modelHash.evaluate(XHash_test, yHash_test, verbose=False)
print("Precisión Prueba:  {:.4f}".format(accuracy))
```

```
Precisión Entrenamiento: 0.9893
Precisión Prueba:  0.9892
```

```
plot_history(historyHash)
```

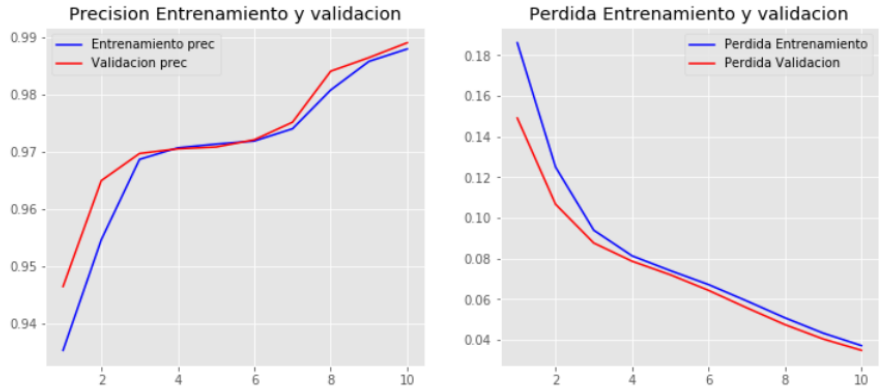


Ilustración 41 - Código Prototipo - Resultados Accuracy y Loss para Red Neuronal II

En esta ocasión vemos un 98% de precisión del modelo, teniendo también sobreajuste, pudiendo por tanto parar el entrenamiento incluso en el *epoch* 3 y teniendo unos resultados muy buenos en la clasificación.

Dado que hemos obtenido estos resultados con valores estándares y topologías sencillas, no vemos relevante, a nivel de resultados a mejorar, el investigar alterando los hiper parámetros, usar regularización, etc… para observar otros resultados posibles.

6 Evaluación

6.1.1 Aplicabilidad:

Este tipo de herramientas implementadas como prototipo, para materializar la idea presentada en este documento, intentarían mejorar la facilidad de catalogación de términos que se encuentren en los sistemas de una corporación, ya sean sistemas gestores de base datos relacionales, no relacionales, datos no estructurados, fuentes *Hadoop Distributed File system (HDFS)* en data lakes, datos en *streaming*, etc… La "des organización" de los activos de datos es uno de los problemas más comunes a los que se enfrenta cualquier organización ya sea empresarial o no, por lo que cualquier utilidad que ayude a enriquecer y alimentar, de una manera totalmente (o semi) automatizada ese diccionario de términos que una organización usa para su negocio, sería de gran valor para la misma.

De esta manera, se aligeran todos los procesos internos en los que el descubrimiento de la naturaleza y semántica de los datos técnicos persistidos es la que más tiempo consume en un proceso de gobierno, conjunto con los procedimientos internos de comprobación, validación y aceptación, de ahí que este tipo de soluciones guiadas por técnicas de inteligencia artificial, sean una realidad plausible a integrar en este tipo de ámbitos de procesos de gobierno de datos.

6.1.2 Encaje en el mapa de arquitectura de la organización

Dentro del esquema de la arquitectura de datos de la organización en cuestión, las piezas de catalogación automática de términos y semántica de los datos técnicos, encajan en un mapa que se muestra en la siguiente vista de implementación y despliegue ("*Implementation and Deployment*" en Archimate):

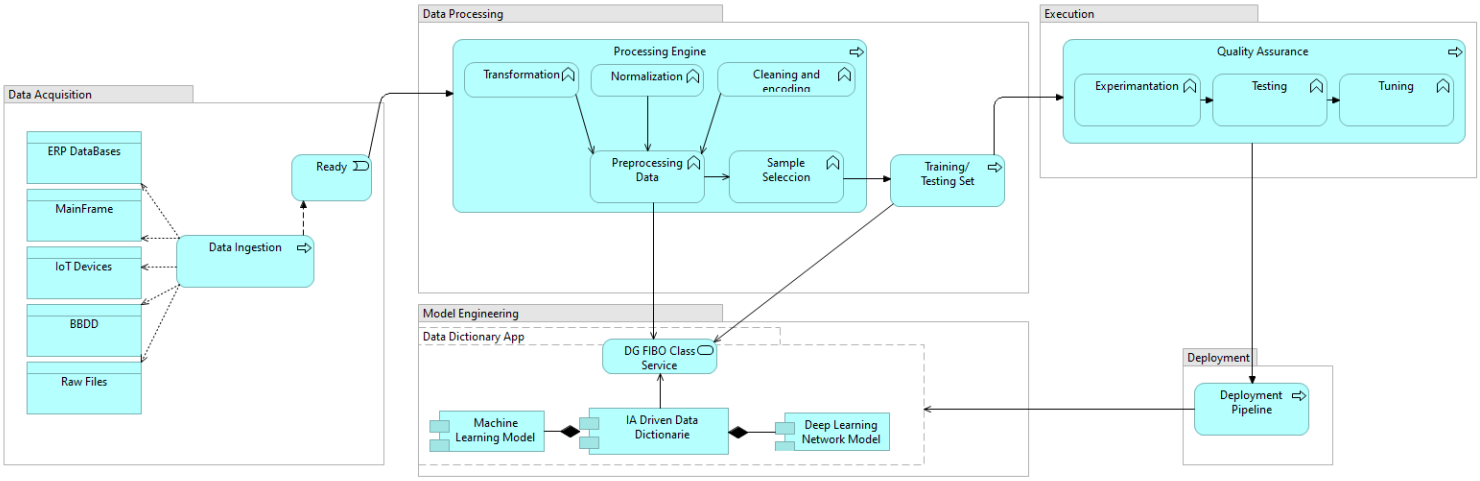


Ilustración 42 - Inclusión del modelo en la Arquitectura de Datos

donde, una vez se haya hecho el procesamiento adecuado (*Data Processing / Preparation*) de los datos provenientes de las diferentes fuentes que se dispongan (*Data Acquisition*), se procedería a desplegar la solución para ser consumida por los diferentes interesados de negocio; por ejemplo, como vemos en la Ilustración 3 para una aplicación transaccional que solicita catalogar los términos de datos técnicos que está manejando, para alimentar el diccionario o catálogo semántico de su negocio.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

En este trabajo presentamos la solución al problema de generación de un catálogo de términos de negocio automatizado usando referencias de ontologías estándares y aplicando técnicas de inteligencia artificial tales como aprendizaje automático, procesamiento de lenguaje natural y sistemas cognitivos artificiales, para ello las aportaciones principales que hemos realizado han sido:

- Hemos llevado a cabo una revisión del estado del arte y hemos observado que todas las aproximaciones y estudios para la asociación de datos técnicos, se están llevando a una línea de emparejamiento directo entre términos de taxonomías y siempre con la necesidad de la participación humana.
    - Productos comerciales como Mastro OBDA
    - Estudios donde se ve la carencia de la correcta catalogación semántica de los datos en entornos Big Data
- Hemos ideado una solución donde el uso de definiciones estándares de lenguajes formales usando ontologías (FIBO en concreto), es clave para la clasificación y enriquecimiento de un glosario de términos en el proceso de gobierno de datos de cualquier organización para el correcto sentido semántico de los datos.
- Hemos planteado el uso de técnicas de Inteligencia artificial para la clasificación de los datos que pueden residir en cualquier soporte y que, a priori, no tienen semántica asociada, para poder enriquecer un catálogo de términos de manera automática, siendo este una pieza fundamental en el proceso de gobierno de datos de las organizaciones para aumentar así, posiblemente, la productividad, tiempos y calidad en el propio proceso.

Tras el estudio de las diferentes aproximaciones en el tratamiento de la información, hemos implementado un prototipo inicial que, a expensas de seguir investigando con un abanico de datos más amplio, da unos buenos resultados en función de los pocos datos que se han podido conseguir en data sets públicos de un dominio muy reducido.

- Hemos experimentado con el tratamiento de los datos asociándolo a técnicas que hemos visto que se están usando para el procesamiento de lenguaje natural como “bolsa de palabras” y tokenización para las caracteristicas.

- Hemos “asociado” registros con “palabras” y “frases” vinculando posteriormente a los datos de entrenamiento y test.

- Hemos diferenciado entre 3 técnicas de inteligencia artificial para la clasificación usando la entidad de la ontología concreta al dato técnico para su clasificación.

    - Clasificador Naive Bayes con vector de caracteristicas tokenizado como datos de entreno y test.

    - Red Neuronal, con matrices dispersas como entrada a la red, generadas a partir de las “frases” del dominio de los datos para su entrenamiento y test

    - Red neuronal, con vector de entrada con codificación numérica tras la transformación de la información de las “frases” a clasificar a un valor numérico, tras aplicar una función hash a los datos textuales.

Dada la naturaleza de los datos que hemos podido conseguir de manera libre y gratuita, los modelos del prototipo implementado, nos dan una fiabilidad superior al 97% en la clasificación de los datos de test:

Naive Bayes	**Red Neuronal** 1 Hidden Layer – 10 neuronas Params: 1.940.947	**Red Neuronal** 1 Hidden Layer – 10 neuronas Params: 567
97,2%	99,79%	98,92%

Tabla 1- Resultados Obtenidos

Los resultados presentan una precisión del 97%, aun teniendo en cuenta un posible porcentaje de *overfitting* (como vemos en las gráficas) pero son prometedores para la realización de las mismas pruebas con un set de datos mucho más amplio para verificar la correcta asignación de entidades de negocio (o Qname) aprendidas de la ontología del

contexto tratado, a una “lista” de atributos técnicos que conforman un registro de información y que tengamos persistidos en nuestra organización.

7.2 Líneas de trabajo futuro

El alineamiento que hemos explicado entre diferentes marcos de arquitectura, ontologías y técnicas de inteligencia artificial hace pensar que se pudiera extender el ámbito de una manera más agregada y homogénea.

Posibles líneas de investigación adicionales:

- **Aproximación “de arriba abajo”**: elegir varios “*Service domain*” del marco de BIAN (para el dominio bancario) donde se explicita qué es lo que realmente comprende una funcionalidad completa, e intentar descubrir via métodos de inteligencia artificial los datos técnicos que residan en un *dataset* determinado. Proceso inverso al que hemos explicado en este trabajo pero ligando el marco BIAN a la Ontología FIBO y así, no sólo enriquecer un diccionario de datos, sino establecer los puentes entre la gestión de datos, la arquitectura funcional y la ontología con semántica, todo ayudado via inteligencia artificial.

  FIBO ←→ BIAN ←→DATO

- **Profundizar en las partículas y sus relaciones:** bajar el nivel de agregación de la aproximación que hemos hecho a nivel de registros, hasta un nivel de atributo y buscar características de proximidad entre atributos para inferir la entidad, o las entidades, de la ontología a la que pertenecería el término para así también incrementar el enriquecimiento usando inteligencia artificial.

- **Inferir términos** que pueden hacer de puntos de unión entre diferentes ontologías y que harían que la web semántica, estuviera correctamente enlazada y que el movimiento *Open linked data* se viera alimentado por algoritmos que generaran las relaciones entre términos, ayudados por modelos de inteligencia artificial.

Ciertamente, creemos que el campo de la gestión de los datos se tiene que ver enriquecido con técnicas de inteligencia artificial, pero siempre haciendo uso de estándares que garanticen el correcto uso, calidad y homogeneidad de la semántica de los datos y que, usar técnicas de inteligencia artificial, es la manera para productivizarlo de una manera

óptima; recordemos que sin datos correctos como entrada, no existe modelo que aprenda ni infiera para una correcta toma de decisiones.

8 Bibliografía

[1] Europea, U. (2016). *Diario oficial de la Unión Europea*. *L 119, 4 de mayo de 2016.*

[2] Conrad, S. S., & Alghamdi, M. (2019). What GDPR means for data privacy. *Journal of Computing Sciences in Colleges*, *34*(3), 133-133.

[3] Alizadeh, F., Jakobi, T., Boldt, J., & Stevens, G. (2019). GDPR-Reality check on the right to access data: claiming and investigating personally identifiable data from companies. In *Proceedings of Mensch und Computer 2019* (pp. 811-814).

[4] Palmér, C. (2017). Modelling EU DIRECTIVE 2016/680 using Enterprise Architecture.

[5] Seabolt, E., Kandogan, E., & Roth, M. (2018, June). Contextual Intelligence for Unified Data Governance. In *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management* (pp. 1-9).

[6] Mosley, M., Brackett, M. H., Earley, S., & Henderson, D. (2010). *DAMA guide to the data management body of knowledge*. Technics Publications.

[7] ISO/IEC. (2019). ISO/IEC 2382-36:2019(en. 2019, de ISO/IEC Sitio web: https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:-36:ed-3:v1:en

[8] Morabito, V. (2015). Big data governance. In *Big data and analytics* (pp. 83-104). Springer, Cham.

[9] López-Cobo, J. M., Losada, S., Cicurel, L., Bas, J. L., Bellido, S., & Benjamins, R. (2008). Ontology management in e-banking applications. In *Ontology Management* (pp. 229-244). Springer, Boston, MA.

[10] Wang, X., Zhou, X., & Jiang, L. (2008, October). A method of business and IT alignment based on enterprise architecture. In *2008 IEEE International Conference on Service Operations and Logistics, and Informatics* (Vol. 1, pp. 740-745). Ieee.

[11] Berndtsson, M., Forsberg, D., Stein, D., & Svahn, T. (2018). Becoming a data-driven organisation.

[12] Kumaran, S., Liu, R., Dhoolia, P., Heath, T., Nandi, P., & Pinel, F. (2008, October). A restful architecture for service-oriented business process execution. In *2008 IEEE International Conference on e-Business Engineering* (pp. 197-204). IEEE.

[13] World Wide Web Consortium (W3C). (2020). Financial Industry Business Ontology (FIBO). 2020, de EDM Council Sitio web: https://spec.edmcouncil.org/fibo/

[14] Reynolds, D. (2014). The Organization Ontology. W3C Recommendation 16 January 2014.

[15] O'Byrne, D., Brennan, R., & O'Sullivan, D. (2010, March). Implementing the draft W3C semantic sensor network ontology. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 196-201). IEEE.

[16] Klotz, B., Troncy, R., Wilms, D., & Bonnet, C. (2018, October). VSSo: The Vehicle Signal and Attribute Ontology. In *SSN@ ISWC* (pp. 56-63).

[17] Lankhorst, M. M., Proper, H. A., & Jonkers, H. (2009). The architecture of the archimate language. In *Enterprise, business-process and information systems modeling* (pp. 367-380). Springer, Berlin, Heidelberg..

[18] Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). *Foundations of semantic web technologies*. CRC press.

[19] Davies, J., Studer, R., & Warren, P. (Eds.). (2006). *Semantic Web technologies: trends and research in ontology-based systems*. John Wiley & Sons.

[20] Domingue, J., Fensel, D., & Hendler, J. A. (Eds.). (2011). *Handbook of semantic web technologies*. Springer Science & Business Media.

[21] Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, *1*(1), 1-136.

[22] Auer, S. (2011, April). The emerging web of linked data. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (pp. 1-1).

[23] Freitas, A., Curry, E., & O'Riain, S. (2012, March). A distributional approach for terminological semantic search on the linked data web. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 384-391).

[24] Parundekar, R., Knoblock, C. A., & Ambite, J. L. (2010, November). Linking and building ontologies of linked data. In *International Semantic Web Conference* (pp. 598-614). Springer, Berlin, Heidelberg.

[25] Babalou, S., & König-Ries, B. (2020). Towards Building Knowledge by Merging Multiple Ontologies with CoMerger: A Partitioning-based Approach. *arXiv preprint arXiv:2005.02659*.

[26] Soru, T., Marx, E., Valdestilhas, A., Moussallem, D., Publio, G., & Saleem, M. (2020). Where is Linked Data in Question Answering over Linked Data?. *arXiv preprint arXiv:2005.03640*.

[27] Freitas, A., Curry, E., & O'Riain, S. (2012, March). A distributional approach for terminological semantic search on the linked data web. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 384-391).

[28*]* Kirstein, F., Stefanidis, K., Dittwald, B., Dutkowski, S., Urbanek, S., & Hauswirth, M. (2020, May). Piveau: A Large-Scale Open Data Management Platform Based on Semantic Web Technologies. In *European Semantic Web Conference* (pp. 648-664). Springer, Cham.

[29] Prud'hommeaux, E. (2008). SPARQL query language for RDF, W3C recommendation. *http://www. w3. org/TR/rdf-sparql-query/*.

[30] Nadal, S., Romero, O., Abelló, A., Vassiliadis, P., & Vansummeren, S. (2019). An integration-oriented ontology to govern evolution in big data ecosystems. *Information systems*, *79*, 3-19..

[31] Haase, P., Broekstra, J., Eberhart, A., & Volz, R. (2004, November). A comparison of RDF query languages. In *International Semantic Web Conference* (pp. 502-517). Springer, Berlin, Heidelberg.

[32] Rodriguez-Muro, M., Kontchakov, R., & Zakharyaschev, M. (2013, October). Ontology-based data access: Ontop of databases. In *International Semantic Web Conference* (pp. 558-573). Springer, Berlin, Heidelberg.

[33] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., ... & Savo, D. F. (2011). The MASTRO system for ontology-based data access. *Semantic Web*, *2*(1), 43-53.

[34] Lai, P., Phan, N., Hu, H., Badeti, A., Newman, D., & Dou, D. (2020). Ontology-based Interpretable Machine Learning for Textual Data. *arXiv preprint arXiv:2004.00204*.

[35] Molnar, C. (2020). *Interpretable Machine Learning*. Lulu. com.

[36] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159-190.)

[37] Munappy, A., Bosch, J., Olsson, H. H., Arpteg, A., & Brinne, B. (2019, August). Data Management Challenges for Deep Learning. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 140-147). IEEE.

[38] Burgdorf, A., Pomp, A., & Meisen, T. (2020). Towards NLP-supported Semantic Data Management. *arXiv preprint arXiv:2005.06916*.

[39] Lenzerini, M. (2011, October). Ontology-based data management. In *Proceedings of the 20th ACM international conference on Information and knowledge management* (pp. 5-6).

[40] Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, *44*(4), 1-36.

[41] Bruggink, D., Karsten, P., & de Meijer, C. (2012). The European cards environment and ISO 20022. *Journal of Payments Strategy & Systems*, *6*(1), 80-99.

[42] BIAN Working groups. (2008). BIAN Service Landscape SL 9.0. 2020, de BIAN - Banking Industry Architecture Network Sitio web: https://www.bian.org/deliverables/the-bian-service-landscape-timeline/

[43] Al-Fedaghi, S., & Alsulaimi, M. (2018, March). Reconceptualization of IT services in Banking Industry Architecture Network. In *2018 7th International Conference on Industrial Technology and Management (ICITM)* (pp. 330-338). IEEE.

[44] Capgemini, The Open Group. (2017). OPEN BUSINESS DATA LAKE (O-BDL) CONCEPTUAL FRAMEWORK. 2017, de The Open Group Sitio web: https://publications.opengroup.org/c172

[45] Liz Petrocelli. (2020). Retail Banking Demo Data. 2020, de data.world Sitio web: https://data.world/lpetrocelli/retail-banking-demo-data

[46] Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protégé-2000: an open-source ontology-development and knowledge-acquisition environment. In *AMIA... Annual Symposium proceedings. AMIA Symposium* (pp. 953-953).

[47] Byrum, J. D. (1999). ISO 639-1 and ISO 639-2: International Standards for Language Codes. ISO 15924: International Standard for Names of Scripts.

[48] Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29-39). London, UK: Springer-Verlag.

[49] Zak Cole. (2018). Data Governance Readiness: The Five Pillars. 2018, de Cloud Data Architect Sitio web: http://www.dataarchitect.cloud/data-governance-readiness-the-five-pillars/

Anexos

Anexo. Artículo de investigación

IA driven Data Governance usando ontologías de dominio

Marcos García Padrón

Universidad Internacional de la Rioja, Logroño (España)

02/09/2020

Resumen

Se demuestra cómo se puede generar un glosario de términos de negocio vinculado a los datos técnicos persistidos en organizaciones para ser usado dentro del proceso de gobierno de datos en las mismas. Usamos técnicas de inteligencia artificial, apoyándonos en ontologías estándares de un dominio concreto, en este caso del dominio bancario como ejemplo aplicado. Se advierte que la clasificación manual es la tendencia más usada queriendo entonces demostrar que la automatización de este proceso es totalmente factible, usando técnicas de clasificación de aprendizaje automático y aprendizaje profundo. Para ello se dispone y se prepara un conjunto de datos ficticio consiguiendo la automatización de dicha clasificación con un nivel de precisión superior al 97% para conjuntos de prueba, concluyendo que la gestión de los datos se puede ver perfectamente enriquecidas con técnicas de inteligencia artificial y así lograr una mayor productividad y eficiencias en las organizaciones eliminando los procesos manuales.

Palabras Clave

Gestión de datos, gobierno de datos, ontologías, inteligencia artificial, aprendizaje profundo.

I. Introducción

Usar diferentes áreas de conocimiento multidisciplinares como la gestión y gobierno de los datos, la inteligencia artificial haciendo uso de estándares internacionales de ontologías de dominio, es el hilo conductor para este trabajo donde, mediante guías de cuerpos de conocimiento y, realizando técnicas de clasificación sobre un conjunto de datos bancarios, conseguiremos demostrar que, el uso de estas técnicas, uniéndolas con las de procesamiento del lenguaje natural, es un campo donde actualmente se está avanzando para conseguir la automatización y mejor productividad en los procesos internos de las organizaciones.

Abordamos la implementación de un prototipo de donde, los resultados obtenidos, nos hacen pensar que el camino de esta investigación es el correcto para poder profundizar más y, poder entrenar los modelos aquí presentados, sobre un conjunto de datos reales y así determinar la idoneidad de la idea de la generación automatizada de un catálogo de negocio corporativo alimentado y descubierto, a través de los datos técnicos persistidos en las diferentes fuentes de datos de las que se disponga en la organización.

II. Estado del Arte

Actualmente el campo de la gestión de los datos ha llegado a un punto en el que las organizaciones se han dado cuenta que sin calidad y sin gobierno de estos, cualquier procesamiento posterior que se quiera tener a nivel de uso de analítica avanzada no acabará en buen fin. Movimientos como la web semántica [1, 2, 3] y *Open Linked data* [4, 5, 6] lo que intentan conseguir es dotar de conocimiento a la web y para ello, una pieza fundamental son las ontologías de dominio, y por ende el gobierno de los datos es una pieza básica, y dentro de este proceso la generación del catálogo de términos que dota de vocabulario de negocio uniéndolos a los datos técnicos. La idea es que ese conocimiento vaya creciendo en base a relaciones de clases de ontologías como se indica en [7] para así enriquecer la información del dominio concreto del que se desea tener conocimiento formal.

En todo el estudio hecho referente al estado del arte, todas las líneas que se han estado llevando a cabo, siempre parten de premisas estáticas de vinculación entre términos de conceptos de negocio con los datos técnicos en sí, pero siempre llevadas a cabo por agentes humanos, así lo podemos ver con aproximaciones como las Piveu [8] o Mastro [9], habiendo esta empezado como estudio de investigación con "*Ontology-Based Data Access (OBDA)"*[10] y acabando siendo un producto comercial. Estas hacen uso de lenguajes de consulta de datos como SPARQL sobre RDF [11] para extraer la información de las entidades, clases, relaciones… es decir la semántica estática inherente en la propia ontología, pero pocos estudios se aproximan a usar técnicas de inteligencia artificial para hacer ese tipo de vinculación de una manera automática tras un proceso de aprendizaje automático o profundo, y así enriquecer de manera automática los conceptos que en ellas se intentan describir. Sí se establecen las bases fundacionales de la gestión de datos con ontologías como puede ser *"Ontology-Based Data Management (OBDM)"[12]* apoyándose igualmente en cuerpos de conocimiento de gestión de datos como DMBok [13], tomándolo como base para construir arquitecturas que incluyan todas las piezas necesarias para esa gestión de datos, y por tanto de su gobierno y glosario de términos.

Recientemente se ve como el uso de técnicas de aprendizaje automático mezclado con el uso de ontologías, prácticamente el

campo que hemos tratado en este trabajo, está en auge. Así apreciamos en [14] el término "*Ontology-based IML (OnML)*" para generar contenido semántico integrando conocimientos específicos de dominio codificado en ontologías, y una extracción de información usando técnicas de *"Interpretable Machine Learning (IML)"* [15].

En [16] se especifica la fase temprana de la investigación que se está llevando a cabo al respecto de este campo y añade una futura línea complementaria con idea de usar procesamiento del lenguaje natural (*NLP Natural Language Processing*) para dotar de otra dimensión a este campo, yendo más allá del valor y las etiquetas de los datos y enriquecerlo con métodos para el aprendizaje de ontologías desde el propio texto [17].

En definitiva, este campo de investigación está en fase muy temprana de obtener resultados en donde se vea realmente el éxito de estas nuevas técnicas de gestión de los datos usando aprendizaje automático, mapeo de ontologías y en última instancia, enriqueciéndolo haciendo uso de lenguaje natural para así, no digamos meta datar, sino de dotar de significado completo de los textos de un ecosistema empresarial real para el gobierno de los datos.

III. Objetivos y Metodología

El principal objetivo de nuestro TFM, es demostrar que se puede construir un catálogo de términos de negocio a partir de datos técnicos persistidos en diferentes fuentes, y crear la vinculación de naturaleza técnica con su significado semántico de una manera automatizada mediante técnicas aprendidas de inteligencia artificial y así enriquecer un proceso de gobierno de datos corporativo.

Plantearemos el uso de técnicas de aprendizaje automático y profundo para eliminar y evitar soluciones estáticas y manuales como las que hemos visto en el apartado del estado del arte, siendo sustituidas estas tareas, por un sistema artificial modelado que prescinda de las tareas del administrador de datos e ingeniero de configuración de mapeos de la ecuación, y aprovechándonos de los estándares mundiales públicos como la ontología *Financial Industry Business Ontology (FIBO)*[18] para reglamentar cualquier nuevo dato.

Para conseguir tales objetivos, estudiaremos y analizaremos un conjunto de datos públicos descargados de la web, asociados al negocio bancario, para intentar relacionar los datos provistos con su correspondiente información semántica identificada en *FIBO*. Una vez tengamos esa asociación, generaremos un prototipo software que intente clasificar y aprender nuevos términos persistidos en las fuentes de datos, hacia su correspondiente entidad de la ontología en cuestión, usando diferentes técnicas de inteligencia artificial.

Para ello, a nivel de desarrollo del propio prototipo software, aplicamos la metodología CRISP-DM [19] y plateamos el piloto con código Phyton en Jupyter notebooks siendo las herramientas que hemos usado durante este master, al igual que aplicamos las técnicas aprendidas de diferntes materias como aprendizaje automático, procesamiento de lenguaje natural y sistemas cognitivos artificiales.

Del mismo modo que la metodología llevada a cabo en el desarrollo del software en sí, debemos mencionar que el éxito de este tipo de aproximaciones también radica en saber "acoplar" la solución software dentro del proceso de gobierno de datos en la organización; para ello hacemos una aproximación de proceso de gobierno de datos y en concreto, de la generación del catálogo de términos, y vemos como la incorporación de este tipo de soluciones aligeraría enormemente muchos de los pasos metodológicos que se suelen tener en este tipo de procesos de gobernanza y catalogación correcta de los datos.

IV. Contribución

La contribución con este trabajo radica en asociar diferentes ideas y experiencias para conseguir solucionar una necesidad que, actualmente, es un problema en cualquier organización.

Vincular los planos lógicos de arquitecturas físicas (persistencia de los datos) con arquitecturas de información (arquitectura de aplicaciones y datos) con ontologías de dominio específico (arquitectura de negocio) y todo ello usando técnicas de inteligencia artificial como aprendizaje automático (con una clasificador "sencillo" como el Naive Bayes) y aprendizaje profundo, ideando un tratamiento de datos concreto para poder afrontar el problema, creemos que realmente es una contribución que aporta una nueva forma de abordar la solución para la necesidad de generación automática de un catálogo de términos de negocio y su trazabilidad con el dato técnico concreto.

Igualmente, creemos que incorporar conceptos aprendidos de procesamiento de lenguaje natural e intentar plasmarlos en la solución, como hemos visto en el estado del arte, es la nueva línea que se está llevando a cabo en el mundo de la investigación para la correcta clasificación y tratamiento de textos, y por ello nuestro trabajo puede contribuir a generar nuevas ideas y líneas de investigación más detalladas y específicas.

V. Evaluación y Resultados

Siguiendo la metodología de desarrollo que hemos aplicado para el prototipo, primeramente, hemos tenido que entender y tratar los datos con los cuales íbamos a trabajar. Hacemos un estudio y entendimiento de *FIBO* que en última instancia son sus entidades las que queremos "aprender" en base a los datos de entrada que tengamos.

Como entrada para nuestro problema, recurrimos a un conjunto de datos público [20] de un banco del sector minorista de la republica checha. Evaluamos internamente los datos provistos y comprobamos la correspondencia entre esos datos financieros con la entidad de la ontología de *FIBO* que corresponde a cada uno de los posibles conceptos que intentan representar dichos datos. Planteamos usar un nivel de agregación de entidad de tabla física a entidad de la ontología (FIBO) para poder llevar a cabo la etapa de modelado de la solución. El nivel de agregación ha sido con el que hemos podido trabajar en función de la cantidad de datos que hemos podido conseguir, teniendo en cuenta que con un mayor detalle en la definición de los datos, se podría haber especificado a nivel más concreto las relaciones entre los términos más atómicos.

Nombre de Dataset	FIBO	Qname
Accounts.csv 4,500 rows, 4 columns	Financial Business and Commerce	fibo-fbc-mod:FBCDomain fibo-fbc-pas-mod:FBCProductsAndServicesModule fibo-fbc-pas-caa:
Client.csv 5,369 rows, 3 columns	Foundations Products and Services Products and Services Ontology Client	fibo-fnd-mod:FNDDomain fibo-fnd-pas-mod:ProductsAndServicesModule fibo-fnd-pas-pas: fibo-fnd-pas-pas:Client

Fig. 1. Ejemplo de asociación datos físicos – entidad FIBO – Qname

Para el preprocesamiento de los datos hemos seguido dos estrategias en función de los métodos de modelado que usamos. Estas radican en dos ideas bastante sencillas con las que hemos abordado nuestro problema:

1. Tener un vocabulario originado por la consecución de las partículas de los datos de entrada, formando con ellos una estructura que "simbolice" frases textuales para aplicar las técnicas de procesamiento de lenguaje natural como "*Bag of words"* y tokenización de términos.
2. Pensar que la entrada del texto, se pudiera asemejar a un valor numérico de la misma manera que se hace para técnicas de visión por ordenador, donde un valor numérico corresponde al nivel de gris en una imagen de blanco y negro, o a diferentes valores de los canales diferenciados que se tienen en una imagen a color.

Una vez hemos considerado esta forma de datos de entrada para nuestros modelos, pasamos a aplicar diferentes algoritmos de clasificación para llevar a cabo el aprendizaje de los mismos.

Evaluación 1- Aprendizaje Automático

La primera aproximación que hemos llevado a cabo para la clasificación es el uso de un clasificador Naive Bayes, donde tras el pre procesamiento de los datos de entrada que hemos identificado anteriormente en el punto 1, efectuaremos la clasificación con este algoritmo usando las utilidades de la librerías provistas como *sklearn*.

Una vez adaptados los datos de entrada y efectuado el aprendizaje de tal clasificador, obtenemos los siguientes resultados en la predicción representándolo en una matriz de confusión:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[ 757    0    0    0    0    0  350]
 [ 116    0    0    0    0    0  113]
 [   0    0 1369    0    0    0    0]
 [   0    0    0 1313    0    0    0]
 [ 100    0    0    0    0    0   70]
 [   0    0    0    0    0 1595  104]
 [   0    0    0    0    0    0 25553]]
```

```
#Confirmamos la confianza del modelo usando metricas proporcionadas por las librerias
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9728689567430026
```

Fig. 2. Matriz de Confusión para un clasificador Naive Bayes

Observamos que obtenemos una confianza del **97%** usando esta técnica "sencilla" de clasificación, por lo que la aproximación que hemos llevado a cabo con estos datos tratados de esta forma, es más que sobresaliente para la resolución del problema que hemos acotado. Observamos igualmente los falsos positivos acontecidos en la matriz de confusión, pudiendo derivar en un estudio más concreto y pormenorizado a posteriori, estudiando el por qué de estos errores y así poder identificar incluso semejanzas semánticas en las clases que están relacionadas en esa predicción.

Evaluación 2 – Aprendizaje profundo

Tras las clasificación llevada a cabo mediante aprendizaje automático, queremos ver otras aproximaciones en la clasificación con otras técnicas, en este caso con aprendizaje profundo.

Para aplicar técnicas de entrenamiento de redes neuronales artificiales, nos basamos en dos aproximaciones diferenciadas principalmente en el cómo interpretamos los datos en la capa de entrada a ala red. Por ellos seguimos dos aproximaciones:

1. Mismo uso de las caracteristicas de los conjunto de datos de entreno y test que el usado para la clasificación Naive Bayes, esto es tokenización de los vectores de entrada.
2. Transformación del conjunto de entrada a vectores numéricos tras aplicar una transformación a tales datos (función hash)

Evaluación 2.1

Usando el mismo conjunto de entrenamiento y test que en la técnica anterior, creamos una red sencilla con una única **capa oculta** con **10 neuronas** conectadas con su capa de entrada y de salida. Dicha topología de red, teniendo en cuenta las **7 clases de salida a clasificar**, que corresponderían con las entidades que hemos identificado con los datos hacia la vinculación de las clases de FIBO, nos generan una red con **1.940.947 parámetros** a entrenar.

Tras el entrenamiento de dicha red, asignando valores de hiper parámetros estándar como puede ser el optimizador *'adam*' y una función de pérdida *'binary_crossentropy'*, conseguimos unos resultados de un 99% de fiabilidad en la clasificación, pero eso sí, a expensas de tener un nivel computacional alto ya que para únicamente **5** ***epochs*** de entrenamiento, el tiempo invertido para el mismo asciende a **20 minutos**.

Vemos que se tiene un gran sobreajuste del modelo, por lo que el disponer de un set de datos de prueba considerable dentro de las clases que queremos clasificar, es un factor primordial para evitar ese sobre ajuste.

Fig. 3. Precisión alcanzada por el modelo tras 5 *epochs*

Fig. 4. Evolución de la función de pérdida del modelo tras 5 *epochs*

Evaluación 2.2

Adicionalmente a la evaluación explicada anteriormente, se plantea una transformación de los datos de entrada para usar una red profunda, aplicando otra variante para la clasificación. Para ello lo que se ha realizado es hacer una aproximación como se indica en el punto V, esto es, modificando los vectores de entrada tokenizados para el entrenamiento, en vectores numéricos aplicando una función hash para tal transformación.

La idea ha sido como se puntualiza con anterioridad, seguir una similitud de técnicas y procedimientos que generalmente se suelen usar en la rama de aprendizaje y reconocimiento de imágenes. El hecho de usar este tipo de aproximaciones de transformación en los datos de entrada radica en la reducción de parámetros para la red y por ende, unos mejores tiempos de ejecución de entrenamiento de la red. Así pasamos de una red de 1.940.947 parámetros a una red (misma topología con una capa oculta de 10 neuronas totalmente conectados con las capas de entrada y salida) **de 567 parámetros**.

Para las ejecuciones con esta técnica de datos de entrada, y mismos valores en los hiper parámetros que en el caso anterior, obtenemos un nivel de **precisión del 98%** y en un tiempo de aprendizaje de **23 segundos**, inclusive doblando el número de *epochs* de entrenamiento (**10 *epochs***).

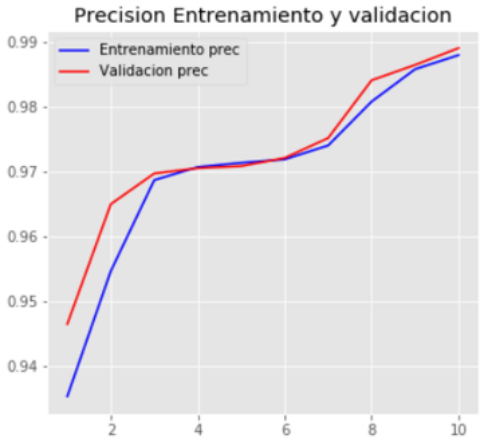


Fig. 5. Precisión alcanzada por el modelo tras 10 *epochs*

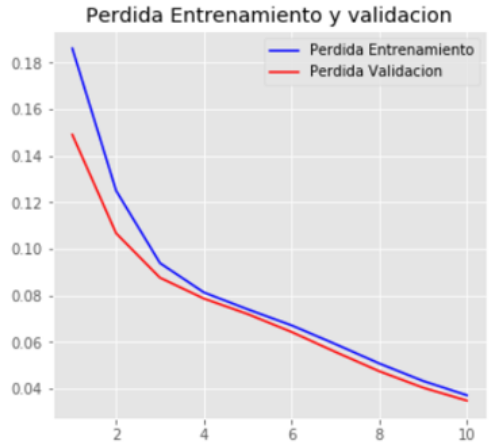


Fig. 6. Evolución de la función de pérdida del modelo tras 10 *epochs*

Observamos un alto sobreajuste en el aprendizaje con respecto a la validación del modelo, del mismo modo que el modelo anterior pudiendo igualmente utilizar parada temprana a partir de la etapa número 4 y minimizar así el tiempo de ejecución y fomentar la generalidad del aprendizaje.

Finalmente, a nivel comparativo teniendo en cuenta los factores de precisión, perdida y tiempo de ejecución vemos que esta ultima forma a de tratar los datos para el aprendizaje sería la más conveniente tras los resultados obtenidos.

VI. Discusión

Tras el análisis de los resultados obtenidos en este trabajo, aplicando las diferentes técnicas de clasificación que hemos expuesto, vemos que técnicas de aprendizaje profundo conjunto con las de procesamiento de lenguaje natural, son las que se decantan como una de las mejores aproximaciones dada la generalidad que se puede llegar a conseguir para que, en base a nuevos términos que tengamos que clasificar en nuestra arquitectura de datos, se pueda clasificar correctamente con un alto nivel de precisión con el correcto termino semántico.

Está claro que en este estudio los datos obtenidos de entrada nos han valido para una primera aproximación relajada de solución, siendo realmente necesario encontrar un conjunto de datos lo suficientemente extenso y completo como para poder verificar con mayor rigor la correcta asignación de clases de la ontología a los registros que podemos tener en nuestro ecosistema de datos de las organizaciones.

Cabe destacar que no sólo la implementación y despliegue técnico de los modelos de clasificación aquí vistos, es el objetivo del trabajo, sino que también la adecuación de los procesos de la organización para contemplar este tipo de soluciones para conseguir una mayor productividad y eficiencia usando técnicas de Inteligencia Artificial. Todo ello es un arduo trabajo a desempeñar en el despliegue de este tipo de soluciones relacionadas con la gestión de los datos, el gobierno de los mismos y la generación del diccionario de conceptos de negocio organizativo.

VII. CONCLUSIONES

En este trabajo presentamos la solución al problema de generación de un catálogo de términos de negocio automatizado usando referencias de ontologías estándares y aplicando técnicas de inteligencia artificial tales como aprendizaje automático, procesamiento de lenguaje natural y sistemas cognitivos artificiales, para ello las aportaciones principales que hemos realizado han sido:

1. Hemos visto que todas las aproximaciones y estudios para la asociación de datos técnicos, se están llevando a una línea de emparejamiento directo entre términos de taxonomías y siempre con la necesidad de la participación humana.

2. Hemos ideado una solución donde el uso de definiciones estándares de lenguajes formales usando ontologías (FIBO en concreto), es clave para la clasificación y enriquecimiento de un glosario de términos en el proceso de gobierno de datos de cualquier organización para el correcto sentido semántico de los datos.

3. Hemos planteado el uso de técnicas de Inteligencia artificial de clasificación para así poder enriquecer un catálogo de términos de manera automática.

Tras el estudio de las diferentes aproximaciones en el tratamiento de la información, hemos implementado un prototipo inicial que, a expensas de seguir investigando con un abanico de datos más amplio, da unos muy buenos resultados en función de los pocos datos que se han podido conseguir en data sets públicos de un dominio muy reducido.

1. Hemos experimentado con el tratamiento de los datos asociándolo a técnicas que hemos visto que se están usando para el procesamiento de lenguaje natural como “bolsa de palabras” y tokenización para las caracteristicas.

2. Hemos “asociado” registros con “palabras” y “frases” vinculando posteriormente a los datos de entrenamiento y test.

3. Hemos diferenciado entre 3 técnicas de inteligencia artificial para la clasificación usando la entidad de la ontología concreta al dato técnico para su clasificación.

Dada la naturaleza de los datos que hemos podido conseguir de manera libre y gratuita, los modelos del prototipo implementado, nos dan una fiabilidad superior al 97% en la clasificación de los datos de test.

Naive Bayes	**Red Neuronal** 1 capa oculta – 10 neuronas Parámetros: 1.940.947	**Red Neuronal** 1 capa oculta – 10 neuronas Parámetros: 567
97,2%	99,79%	98,92%

Está claro que estos resultados son demasiado “buenos” (resultando en sobreajuste) pero son prometedores para la realización de las mismas pruebas con un set de datos mucho más amplio y así verificar la correcta asignación de entidades de negocio (o Qname) aprendidas de la ontología del contexto tratado, a una “lista” de atributos técnicos que conforman un registro de información y que tengamos persistidos en nuestra organización.

Líneas de trabajo futuro

El alineamiento que hemos explicado entre diferentes frameworks de arquitectura, ontologías y técnicas de inteligencia artificial hace pensar que se pudiera extender el ámbito de una manera más agregada y homogénea.

Posibles siguientes líneas de investigación adicionales:

- **Aproximación “de arriba abajo”:** Empezar la clasificación desde los conceptos de negocio hacia los datos técnicos.
- **Bajar el nivel de agregación** para buscar características de proximidad entre atributos e inferir la entidad de la ontología.
- **Inferir términos** que pueden hacer de puntos **de unión entre diferentes ontologías** enriqueciendo así la web semántica.

Ciertamente, creemos que el campo de la gestión de los datos se tiene que ver enriquecido con técnicas de inteligencia artificial, pero siempre haciendo uso de estándares que garanticen el

correcto uso, calidad y homogeneidad de la semántica de los datos y que, usar técnicas de inteligencia artificial, es la manera para productivizarlo de una manera óptima; recordemos que sin datos correctos como entrada, no existe modelo que aprenda ni infiera para una correcta toma de decisiones.

APÉNDICES

REFERENCIAS

[1] Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). Foundations of semantic web technologies. CRC press.

[2] Davies, J., Studer, R., & Warren, P. (Eds.). (2006). Semantic Web technologies: trends and research in ontology-based systems. John Wiley & Sons.

[3] Domingue, J., Fensel, D., & Hendler, J. A. (Eds.). (2011). Handbook of semantic web technologies. Springer Science & Business Media.

[4] Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology, 1(1), 1-136.

[5] Auer, S. (2011, April). The emerging web of linked data. In Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications (pp. 1-1).

[6] Freitas, A., Curry, E., & O'Riain, S. (2012, March). A distributional approach for terminological semantic search on the linked data web. In Proceedings of the 27th Annual ACM Symposium on Applied Computing (pp. 384-391).

[7] Babalou, S., & König-Ries, B. (2020). Towards Building Knowledge by Merging Multiple Ontologies with CoMerger: A Partitioning-based Approach. arXiv preprint arXiv:2005.02659.

[8] Kirstein, F., Stefanidis, K., Dittwald, B., Dutkowski, S., Urbanek, S., & Hauswirth, M. (2020, May). Piveau: A Large-Scale Open Data Management Platform Based on Semantic Web Technologies. In European Semantic Web Conference (pp. 648-664). Springer, Cham.

[9] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., ... & Savo, D. F. (2011). The MASTRO system for ontology-based data access. Semantic Web, 2(1), 43-53.

[10] Rodriguez-Muro, M., Kontchakov, R., & Zakharyaschev, M. (2013, October). Ontology-based data access: Ontop of databases. In International Semantic Web Conference (pp. 558-573). Springer, Berlin, Heidelberg

[11] Haase, P., Broekstra, J., Eberhart, A., & Volz, R. (2004, November). A comparison of RDF query languages. In International Semantic Web Conference (pp. 502-517). Springer, Berlin, Heidelberg.

[12 Lenzerini, M. (2011, October). Ontology-based data management. In Proceedings of the 20th ACM international conference on Information and knowledge management (pp. 5-6).

[13] Mosley, M., Brackett, M. H., Earley, S., & Henderson, D. (2010). DAMA guide to the data management body of knowledge. Technics Publications.

[14] Lai, P., Phan, N., Hu, H., Badeti, A., Newman, D., & Dou, D. (2020). Ontology-based Interpretable Machine Learning for Textual Data. arXiv preprint arXiv:2004.00204.

[15] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3), 159-190.

[16] Burgdorf, A., Pomp, A., & Meisen, T. (2020). Towards NLP-supported Semantic Data Management. arXiv preprint arXiv:2005.06916

[17] Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. ACM Computing Surveys (CSUR), 44(4), 1-36

[18] World Wide Web Consortium (W3C). (2020). Financial Industry Business Ontology (FIBO). 2020, de EDM Council Sitio web: https://spec.edmcouncil.org/fibo

[19] Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (pp. 29-39). London, UK: Springer-Verlag.

[20] Liz Petrocelli. (2020). Retail Banking Demo Data. 2020, de data.world Sitio web: https://data.world/lpetrocelli/retail-banking-demo-data