

E4.3 MÉTODOS DE EXTRACCIÓN DE GRAFOS DE CONOCIMIENTO A PARTIR DE TEXTO MEDIANTE LLM

KG4LLM - SERVICIOS PARA EL ENRIQUECIMIENTO DE MODELOS DE LENGUAJE
CON GRAFOS DE CONOCIMIENTO
(SER-21/23 OTT)

Resumen

Este entregable reporta el proceso seguido para ajustar modelos de lenguaje en la tarea de extracción de instancias de una ontología a partir de texto. El objetivo es generar modelos especialistas en la tarea de poblar ontologías utilizando Turtle como serialización RDF. Se experimenta en el dominio de las misiones espaciales dado que existe una ontología, descripciones de texto y datos estructurados de las misiones que son insumos necesarios para crear, con la ayuda de un LLM del estado del arte, un conjunto de datos de entrenamiento y evaluación. Los resultados de la evaluación demuestran que es posible ajustar modelos, con un número reducido de parámetros, especialistas en esta tarea. Este resultado es importante porque los modelos más pequeños tienen, entre otros beneficios, menores requisitos de memoria de GPU lo que reduce el costo de su puesta en producción.

Andres Garcia-Silva
José Manuel Gómez Pérez

30 de diciembre de 2024
Expert.ai Research Lab

Historia de modificaciones

Revisión	Fecha	Descripción	Autor
0.1	18 dic 2024	Primera versión completa del documento	Expert.ai
1.0	30 dic. 2024	Versión final revisada	Expert.ai

Contenido

1	Introducción.....	4
2	Modelos Pythia.....	4
3	Ontología, texto y datos estructurados de misiones espaciales.....	5
3.1	EOPortal: Descripciones de texto de las misiones	5
3.2	Base de datos CEOS: Datos estructurados de las misiones	6
4	Generación de los datos de entrenamiento y validación.	6
5	Ajuste de los modelos de Pythia.....	8
6	Evaluación de los modelos ajustados.....	11
7	Conclusiones.....	15
8	Referencias	16
Apéndice A:	Ontología de misiones espaciales	17

1 Introducción

Poblar ontologías a partir de texto es un proceso complejo que tradicionalmente se ha realizado siguiendo procesos con múltiples puntos de fallo. Un proceso de población típico (Ji y Grishman, 2011) de ontologías inicia con el enlazado de entidades (Wu et al., 2019) para reconocer las entidades mencionadas en el texto, seguido de un proceso de *slot-filling* (Surdeanu y Ji, 2014) para predecir el objeto para cada sujeto y relación nombrada dado el texto donde aparece el sujeto. La tarea de slot-filling suele incluir un componente de extracción de relaciones que identifica la relación entre un sujeto y el objeto candidato.

Dado el avance que los LLM han traído al procesamiento y generación de texto se propone usar un LLM para realizar la tarea de poblar una ontología a partir de un corpus. Esto es, usar un LLM para analizar un texto de entrada, extraer las instancias y generarlas en una serialización de RDF, como puede ser Turtle. Los LLM comerciales como GPT4.0 son capaces de generar código Turtle y de realizar la extracción de instancias. Sin embargo, el foco de este trabajo son modelos mucho más pequeños que por medio de un proceso de ajuste pueden volverse especialistas en la tarea de población de ontologías. Un LLM más pequeño es más eficiente y tiene un consumo de memoria de GPU reducido lo que implica menores costos en la ejecución.

Ajustar un LLM para que aprenda a poblar una ontología usando formato Turtle requiere contar con un conjunto de datos de entrenamiento que incluya tanto los textos desde donde se va a extraer las instancias de la ontología y la correspondiente formalización en formato Turtle de las instancias de las ontologías extraídas del texto. Con el conjunto de datos se puede iniciar el ajuste del LLM en la tarea de modelado de lenguaje. El modelado de lenguaje es la tarea original en la que los LLM han sido entrenados para aprender a predecir el siguiente token en la secuencia de entrada. En este caso la secuencia de entrada son las instrucciones que se le dan al LLM para que extraiga las instancias a partir de un texto de entrada y la salida esperada es el código Turtle de las instancias extraídas.

Para los experimentos reportados en este entregable se ha seleccionado el dominio de las misiones espaciales debido a que existen tanto descripciones de texto como información estructurada de las misiones. Las descripciones de las misiones se usan como texto del cual se extraen las instancias, y la información estructurada es útil para validar la información extraída. Además, se selecciona la familia de modelos Pythia para ajustarlos a la tarea de poblar ontologías en el dominio de las misiones espaciales. Pythia incluye 8 modelos en el rango de los 14M a 12B de parámetros, todos entrenados en el mismo corpus y en condiciones similares, lo que facilita la comparación de los resultados del proceso de ajuste.

Los resultados de los experimentos demuestran que es posible ajustar modelos especialistas en la población de ontologías. El modelo con mejor desempeño tiene 2.8B de parámetros, lo que demuestra que para tareas como esta puede ser contraproducente entrenar modelos más grandes dependiendo del número de muestras en el conjunto de entrenamiento. Incluso un modelo de 1B de parámetros tiene un desempeño comparable a los modelos de 6.9B y 12B de parámetros.

2 Modelos Pythia

La familia Pythia incluye 8 LLM, todos entrenados con datos públicos vistos en exactamente el mismo orden y con un tamaño que va desde 14 millones hasta 12 mil millones de parámetros. Pythia se desarrolló con el propósito explícito de permitir la investigación en interpretabilidad, dinámica de aprendizaje, ética y transparencia. Pythia proporciona acceso público a 154 puntos de control (checkpoints) para cada uno de los modelos, junto con herramientas para descargar y reconstruir sus cargadores de datos de entrenamiento para estudios posteriores.

Los modelos Pythia se entrena en The pile¹ y su versión sin duplicados. The pile es un conjunto de datos de 800 Gigabytes de texto diverso para el modelado de lenguaje. The pile contiene

¹ <https://pile.eleuther.ai>

muchos dominios dispares, incluidos libros, repositorios de GitHub, páginas web, registros de chat y artículos médicos, de física, matemáticas, informática y filosofía. Cada modelo vio aproximadamente 300.000.000 tokens durante el entrenamiento. Esto corresponde a poco menos de 1 época en The pile para los modelos "estándar" y $\sim 1,5$ épocas en The pile sin duplicados (que contiene 207.000.000 tokens en 1 época). Los modelos disponibles se listan en la Tabla 1.

Tabla 1. Modelos Pythia disponibles y características de arquitectura y parametros entrenamiento

Params	n_layers	d_model	n_heads	d_head	Batch Size	Learning Rate
14M	6	128	4	32	2M	1.0e-3
31M	6	256	8	32	2M	1.0e-3
70M	6	512	8	64	2M	1.0e-3
160M	12	768	12	64	2M	6.0e-4
410M	24	1024	16	64	2M	3.0e-4
1B	16	2048	8	256	2M	3.0e-4
1.4B	24	2048	16	128	2M	2.0e-4
2.8B	32	2560	32	80	2M	1.6e-4
6.9B	32	4096	32	128	2M	1.2e-4
12B	36	5120	40	128	2M	1.2e-4

3 Ontología, texto y datos estructurados de misiones espaciales

La ontología de misiones espaciales fue inicialmente descrita en Berquand y Ladeira (2022) (ver Figura 1). Sin embargo, más allá de la representación gráfica la ontología no está disponible en formato RDF/OWL. Por tanto, el primer paso consiste en formalizar la ontología en RDF. Para modelar y formalizar la ontología se usa Protegé. La ontología resultante se presenta en el Apéndice A: Ontología de misiones espaciales.

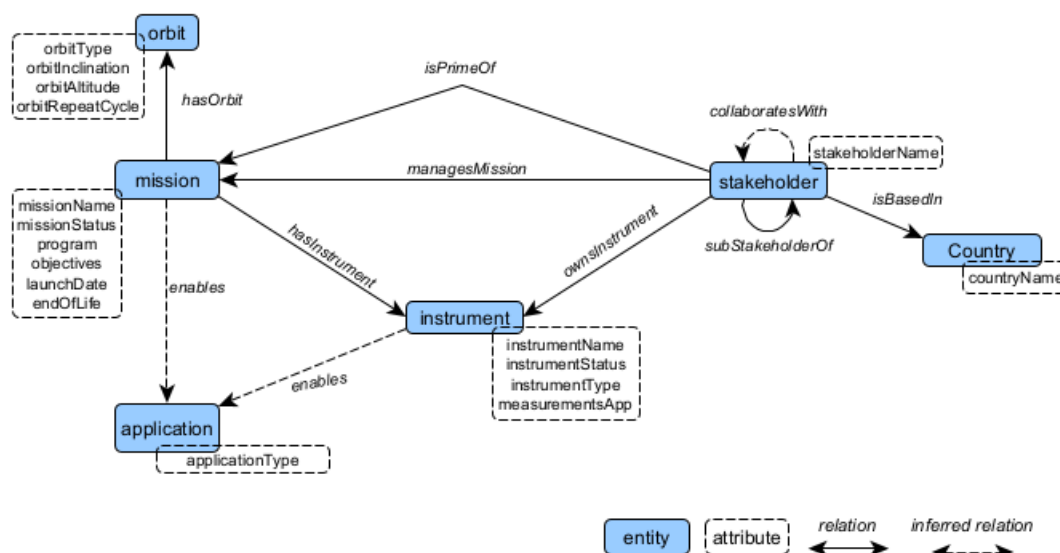


Figura 1. Ontología de misiones espaciales por Berquand y Ladeira (2022). Imagen tomada del artículo "From Mission Description to Knowledge Graph: Applying Transformer-based models to map knowledge from publicly available satellite datasets"

3.1 EOPortal: Descripciones de texto de las misiones

El portal EOPortal (<https://www.eoportal.org>) contiene más de 1.200 artículos que describen misiones espaciales. Cada artículo proporciona una descripción textual de una misión. Aunque las páginas web no siguen ninguna plantilla estándar, usualmente contienen una visión general

de la misión, una descripción del segmento espacial, de los instrumentos y de los segmentos terrestres, así como información sobre el lanzamiento y el estado de la misión.

Para extraer el texto del portal se usa la librería scrapy² de Python. En total se recuperaron 1.069 descripciones de misiones. Cada descripción incluye el título de la misión, el resumen y todo el texto que la describe.

3.2 Base de datos CEOS: Datos estructurados de las misiones

La base de datos de la Comisión de Satélites de Observación de la Tierra (CEOS) proporciona información sobre los satélites basada en una encuesta anual de las agencias miembros del CEOS y representa la única declaración oficial consolidada de los programas y planes de las agencias. La base de datos del CEOS se actualiza anualmente y es el único informe oficial consolidado de los programas y planes de las agencias del CEOS.

La base de datos CEOS, en particular las tablas describiendo las misiones y los instrumentos se puede descargar de la página web en formato Excel. En total se descarga información estructurada de 725 misiones y 1.055 instrumentos. Solo 172 misiones en la base de datos CEOS están descritas en EOPortal.

4 Generación de los datos de entrenamiento y validación.

Para entrenar el modelo es necesario tener un conjunto de datos con las instrucciones para extraer instancias de la ontología, el texto de entrada que describe la misión y la salida esperada, que consiste en las instancias extraídas en formato Turtle. Para generar este conjunto de datos usamos Llama3-8B³, que es un LLM del estado de arte de los modelos de 8B liberado por Meta en abril de 2024.

Llama3-8B tiene un límite de secuencia máximo de 8.192 tokens. En este límite se tiene que incluir las instrucciones, la ontología, el texto que describe la misión, de forma opcional el ejemplo de la tarea realizar, y la salida esperada que en este caso es el código Turtle. La longitud de las descripciones de las misiones de EOPortal es muy extensa y al introducirla en la secuencia con la que se entrena el LLM puede exceder el límite. Por esto, y teniendo en cuenta que los modelos de Pythia que se van a entrenar tienen un límite máximo de secuencia de 2,048 tokens, que es muy inferior se decide resumir las descripciones usando llama3-8B.

Por esto se instruye a llama3-8B para que resuma el texto que describe cada misión en 3 o 4 párrafos centrándose en la información principal de la ontología, es decir, información de la misión incluyendo fecha de lanzamiento y finalización, objetivos, órbita, instrumentos y partes involucradas.

Para generar el conjunto de datos de entrenamiento y validación se dan las siguientes instrucciones a Llama3-8B en forma de diálogo:

- Se le pide al modelo que reconozca que entiende la ontología de misiones que se le presenta a continuación. La ontología se describe en texto (no en RDF)
- El modelo responde que entiende la ontología
- Se le dan instrucciones para que extraiga instancias de la ontología a partir del resumen de la descripción de una misión y se le provee un ejemplo de dicha tarea.
- El modelo responde con las instancias de la ontología que extrae del texto de entrada

La longitud promedio de este prompt es de 2896 tokens. Con este procedimiento se procesan un total de 1.060 descripciones de misiones tomadas de EOportal. La generación se realiza con los parámetros temperature=0.6 y top_p=0.9, valores que se obtuvieron mediante experimentación previa.

² <https://scrapy.org>

³ <https://ai.meta.com/blog/meta-llama-3>

Sin embargo, Llama3-8B no genera Turtle valido sintácticamente para todas las misiones que se le presentan, por lo que se aplican diferentes heurísticas para convertir Turtle no valido en valido. Además, se realizan varias iteraciones de generación para de esta manera intentar generar Turtle valido para la mayoría de las misiones. Las heurísticas aplicadas a los Turtle generados no validos son:

- **Namespaces NS:** Remplazar todos los namespaces en el Turtle generado por los correctos.
- **Correcciones léxico-sintácticas**
 - o **Corrección de formato de fechas:** Adición de hora por defecto a fechas sin hora, eliminación de tripletas que incluyan en el objeto fechas “unknown” o vacías.
 - o **Corrección de entidades que inician con interrogación:** Se elimina la interrogación del inicio de sujetos y objetos.
- **Correcciones Semánticas:**
 - o **Corrección de Dominio:** Se borran todas las tripletas que no cumplan con las restricciones de dominio expresadas en la ontología.
 - o **Corrección de Rango:** Se borran todas las tripletas que no cumplan con las restricciones de rango expresadas en la ontología.
 - o **Relaciones:** Instrumentos y órbitas aisladas se relacionan automáticamente con la entidad que representa la misión.
 - o Remplazar relaciones de **subclase** de a rdf type.

Los resultados de la generación de código Turtle para las instancias extraídas de los resúmenes de las misiones se presenta en la

Tabla 2.

Tabla 2. Estadísticas de porcentaje de Turtle valido y no valido sintácticamente de las rondas de generación de instancias de la ontología de misiones del espacio. Se incluye el resultado de modificar el código adicionado los namespaces correctos (NS) y realizando modificaciones sintácticas y semánticas.

Ronda de Generación	Misiones	Turtle valido				Turtle no valido
		Generación	NS	Otras-lexSint.	Semánticas	
1	1060	243	534	552	544	516
2	516	97	208	221	220	296
3	296	49	88	94	91	205
4	205	26	42	46	46	159
5	159	17	43	50	50	109
6	109	12	23	25	25	84
7	84	14	27	22	27	57
8	57	7	11	13	13	44
9	44	3	8	9	9	35

En total se genera Turtle valido sintácticamente para 1.025 misiones en 9 rondas de generación. El reemplazo de los namespaces es la heurística que mayor impacto tiene a la hora de generar Turtle valido (ver Figura 2).

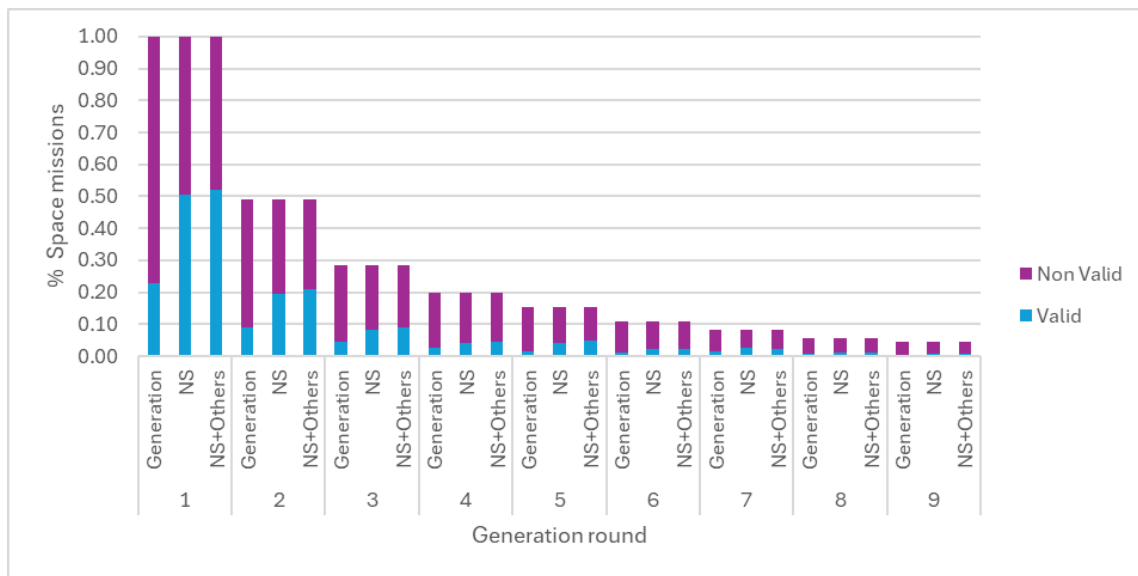


Figura 2. Porcentaje de misiones con Turtle generado valido y no valido sintácticamente. Se incluye el porcentaje de misiones con Turtle valido y no valido despues de realizar las corrección de los namespaces y correcciones sintacticas y semánticas.

5 Ajuste de los modelos de Pythia

El conjunto de datos generado en el apartado anterior se divide en datos de validación que incluyen los datos de las 172 misiones (17% del total) que cuentan con datos estructurados en la base de datos CEOS. Los datos de las 853 misiones restantes (83% del total) se usan para entrenamiento.

Para ajustar los modelos Pythia en la tarea de extraer instancias de la ontología de misiones a partir de texto y formalizarlas en formato Turtle se realiza un entrenamiento adicional en la tarea de modelado de lenguaje. El prompt de cada modelo consiste en:

- Opcionalmente se puede incluir una descripción en modo texto de la ontología
- Las instrucciones para extraer las instancias de la ontología a partir del texto.
- El texto con el resumen de la misión que se ha extraído de EOPortal.

El modelo se entrena para predecir las instancias que se extraen del texto de entrada en formato Turtle. A continuación, se listan los hiper parámetros que se deben definir al entrenar un LLM.

- **Batch Size (B.S.):** representa la cantidad de muestras utilizadas al mismo tiempo para entrenar una red neural y tiene un impacto directo en la precisión y la eficiencia computacional del proceso de entrenamiento. El *batch size* puede entenderse como un equilibrio entre precisión y velocidad. Los *batch size* grandes pueden generar tiempos de entrenamiento más rápidos, pero pueden resultar en una precisión menor y sobreajuste, mientras que los *batch size* más pequeños pueden brindar una mejor precisión, pero pueden ser computacionalmente costosos y consumir mucho tiempo. El *batch size* también puede afectar la convergencia del modelo, es decir, puede influir en el proceso de optimización y la velocidad a la que aprende el modelo. Los *batch size* pequeños pueden ser más susceptibles a fluctuaciones aleatorias en los datos de entrenamiento, mientras que los más grandes son más resistentes a estas fluctuaciones, pero pueden converger más lentamente.
- **Steps:** Se refiere al número de *batches* procesados durante el entrenamiento del modelo. En cada step se entrena el modelo con las muestras en un *batch*.
- **Gradient Accumulation:** Es una técnica que permite entrenar con *Batch Size* más grandes que los que la máquina normalmente podría almacenar en la memoria. Esto se hace acumulando gradientes en varios *batches* y ejecutando el optimizador solo después de que se haya realizado una cierta cantidad de *batches*.

- **Gradient Checkpointing:** Es un método utilizado para reducir la huella de memoria durante el entrenamiento de redes neuronales profundas, a costa de tener un pequeño aumento en el tiempo de cálculo. Esta técnica guarda activaciones seleccionadas estratégicamente en todo el gráfico computacional, por lo que solo es necesario volver a calcular una fracción de las activaciones para los gradientes.
- **Early Stopping:** Detiene el entrenamiento cuando una métrica monitoreada haya dejado de mejorar. La métrica utilizada es la función de validación aplicada al conjunto de validación. Cuando la función de validación deja de mejorar entonces se termina el entrenamiento del modelo.

Para evaluar el entrenamiento de los modelos se usan como métricas:

- **Perplexity:** Es una de las métricas más comunes para evaluar modelos de lenguaje. Es una medida de incertidumbre en el valor de una muestra de una distribución de probabilidad discreta. Cuanto mayor sea esta métrica, menos probable será que un observador pueda adivinar el valor que se extraerá de la distribución. En el caso de los LLM, cuanto mayor sea la métrica de perplexity menos probable será que el LLM pueda adivinar la siguiente palabra en la secuencia de entrada.
- **Validation loss:** La función de pérdida aplicada a los datos de validación es una medida de la eficacia con la que el modelo se generaliza al conjunto de validación. Representa el error en los datos no vistos. Una función de pérdida aplicada a los datos de validación creciente indica que el rendimiento del modelo en el conjunto de validación está empeorando, lo que sugiere que se está volviendo menos eficaz a la hora de generalizar a nuevos datos.

Finalmente, para entrenar modelos grandes de más de 2.8B de parámetros que requieren grandes cantidades de memoria GPU, se utiliza LoRA:

- **Low-rank Adaptation LoRA:** es una técnica para adaptar rápidamente los modelos de aprendizaje automático a nuevos contextos. Funciona añadiendo piezas ligeras al modelo original, en lugar de cambiar el modelo entero. LoRA reduce la cantidad de parámetros entrenables al aprender pares de matrices de descomposición de bajo rango mientras congela los pesos originales. Esto reduce enormemente el requisito de almacenamiento para modelos de lenguaje grandes adaptados a tareas específicas, todo ello sin introducir latencia de inferencia.

En la Tabla 3 se presenta los parámetros de entrenamiento para los modelos Pythia y las métricas de evaluación correspondiente. Los modelos Pythia que se ajustaron estaban entrenados previamente en la versión sin duplicados de The pile si estaban disponibles. Además, en el entrenamiento se usó early stopping = 2 para que en caso de que la función de pérdida no mejorará en dos steps consecutivos se detuviera el entrenamiento. El batch size se definió en 8, y cuando el entrenamiento generaba problemas de memoria se usó Gradient accumulation para mantener el batch size efectivo en 8. Finalmente se ajustaron dos modelos por cada modelo Pythia, uno que incluía la ontología en el prompt y otro que no lo incluía.

Tabla 3. Modelos Pythia entrenados para generar código Turtle con las instancias de la ontología de misiones espaciales que pueden extraer de descripciones de las misiones. En la tabla se incluye los parámetros de entrenamiento cada modelo y métricas de evaluación de los modelos entrenados.

Model	Gradient Checkpointing	LoRA	B. S.	Gradient Accumulation	Ontology in prompt		No ontology in prompt	
					Perplexity	Val. Loss	Perplexity	Val. Loss
14m	No	No	8	0	1.35	0.297	1.35	0.303
31m	No	No	8	0	1.30	0.261	1.30	0.259
70m	No	No	8	0	1.26	0.231	1.25	0.225
160m	Yes	No	8	0	1.19	0.173	1.18	0.167
410m	Yes	No	8	0	1.13	0.123	1.13	0.122

1B	Yes	No	8	0	1.11	0.104	1.11	0.103
1.4B	Yes	No	4	2	1.11	0.107	1.11	0.103
2.8B	Yes	yes	4	2	1.09	0.082	1.09	0.090
6.9B	Yes	yes	1	8	1.08	0.080	1.09	0.083
12B	Yes	Yes	1	8	1.10	0.092	1.09	0.088

En total, los modelos se entrenaron por un máximo de 700 steps, lo que implica que cada modelo vería como máximo aproximadamente 7 veces el conjunto de entrenamiento (106 steps por época). La función de perdida en el conjunto de datos de validación en cada step para las variantes con ontología y sin ontología en el prompt se presenta en las figuras Figura 3 y Figura 4. En estas figuras se puede observar cómo el entrenamiento de los modelos de 1B y 1.4B convergen más rápido que el resto de los modelos. Además, se puede apreciar que los modelos de menos de 160M de parámetros tienen una función de perdida significativamente mayor que los modelos más grandes. En general, los modelos de más de 1B de parámetros tienen una función de perdida significativamente menor que los modelos más pequeños. El modelo que presenta una función de perdida menor en ambos escenarios (con o sin ontología en el prompt) es el modelo de 6.9B.

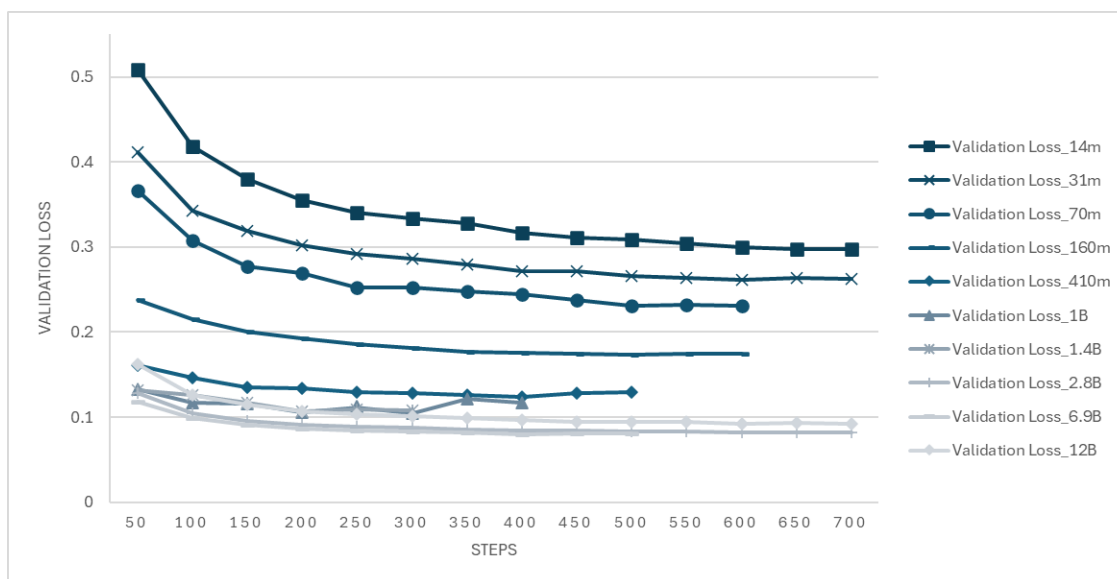


Figura 3. Función de perdida aplicada al conjunto de validación de los modelos Pythia entrenados usando la ontología en el prompt.

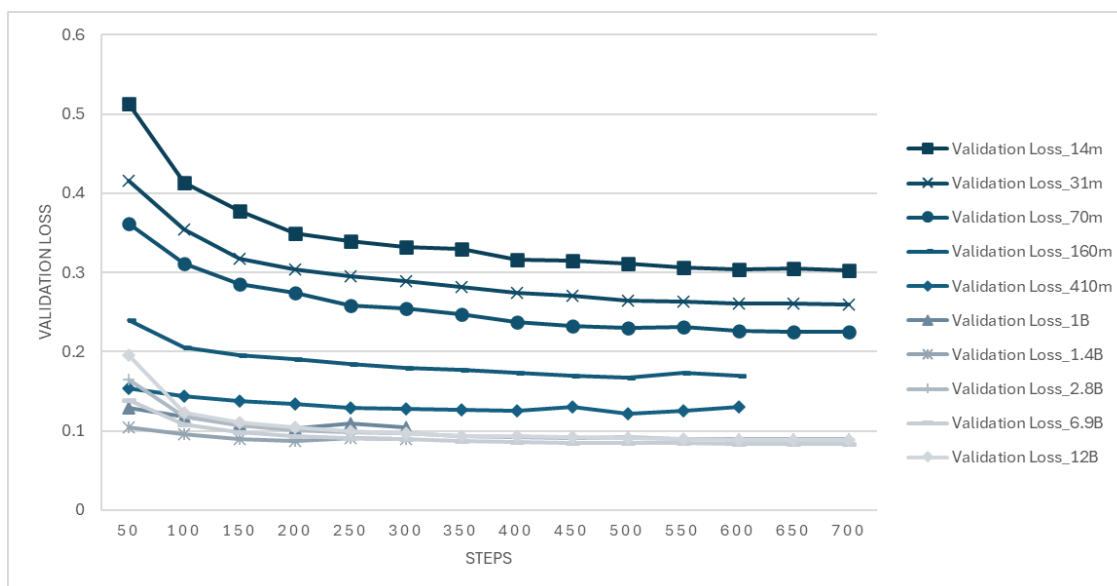


Figura 4. Función de pérdida aplicada al conjunto de validación de los modelos Pythia entrenados sin usar la ontología en el prompt.

6 Evaluación de los modelos ajustados

Aunque la función de pérdida y perplexity sugieren que los modelos de más de 1B de parámetros han aprendido a extraer instancias de los textos de las misiones y formalizarlas en Turtle, es necesario realizar una evaluación que valide que el Turtle extraído es correcto sintácticamente y que la información que se ha extraído de los textos corresponde a las misiones procesadas.

En la Figura 5 se presenta el porcentaje de Turtle válido sintácticamente generado por los modelos ajustados en el conjunto de datos de validación. Desde 160m los modelos generan Turtle válido sintácticamente. Si se compara los modelos Pythia ajustados con llama3-8B en zero-shot, es decir sin ajustarse a la tarea, se observa que los modelos ajustados con 410M de parámetros o más generan un porcentaje mayor de Turtle válido, lo que valida que los modelos ajustados han aprendido la tarea y lo hacen mejor que un modelo de 8B del estado del arte. Además, incluir la ontología en el prompt perjudica a los modelos de 2.8B de parámetros o más, mientras que en los de 1.4B y 1B de parámetros los beneficia.

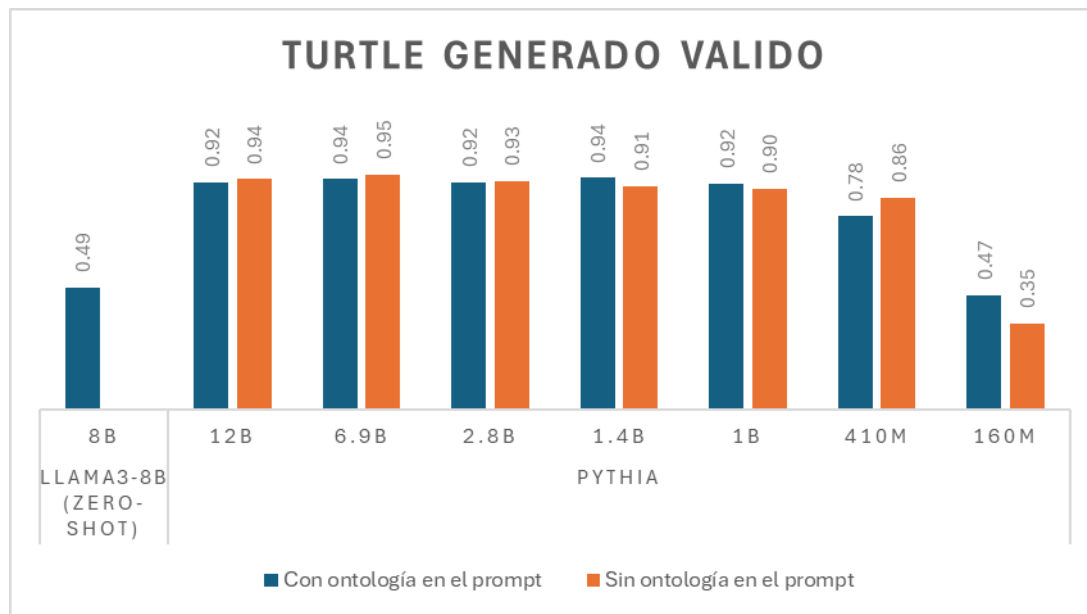


Figura 5. Porcentaje de Turtle valido sintácticamente generado a partir del conjunto de validación por los modelos ajustados. Además, se presenta el resultado de llama3-8B (zero-shot) como línea base.

Para evaluar semánticamente la generación de los modelos ajustados se usa la información de la base de datos CEOS. Cada Turtle generado se consulta usando SPARQL para identificar las propiedades de las instancias de las clases en la ontología de misiones espaciales. El objeto de cada propiedad se compara con el valor de dicha propiedad en la tabla de Misiones o Instrumentos en la base de datos CEOS. Como métrica de comparación se utiliza:

- Rouge-L: ROUGE es un conjunto de métricas usadas para evaluar tareas de resumen o traducción automática, al comparar un resumen o traducción producido automáticamente con un resumen o traducción de referencia o un conjunto de referencias. Las métricas de ROUGE varían entre 0 y 1, y las puntuaciones más altas indican una mayor similitud entre el resumen producido automáticamente y la referencia. ROUGE-L se basa en la subsecuencia común más larga (LCS) entre el resultado generado por el modelo y la referencia, es decir, la secuencia más larga de palabras (no necesariamente consecutivas, pero aún en orden) que comparten ambos. Una secuencia compartida más larga indica una mayor similitud entre las dos secuencias.
- Similitud basada en LLM: Se utiliza un LLM como juez que compare la salida generada por los modelos con el valor de la propiedad en la base de datos CEOS. En este caso se usa Llama3-8B como juez al que se instruye para que compare semánticamente la generación con el valor esperado y genere un puntaje entre 0 y 1 que indique que tan similares son.

En las tablas Tabla 4 y Tabla 5 se presenta los resultados de la evaluación de los modelos ajustados incluyendo o no la ontología en el prompt por cada propiedad evaluada en la ontología. Para ciertas propiedades como “Objetivo de la Mission” o “Aplicaciones de las Medidas de los Instrumentos” Rouge-L genera puntajes muy diferentes y bajos comparados con los del LLM. Esto se debe a que la generación de los LLM ajustados es bastante más concisa que las respuestas esperadas en la base de datos CEOS. La similitud basada en LLM genera puntajes más altos debido a que el LLM considera la semántica de los textos y no se basa tanto en la forma en que estén escritas.

Tabla 4. Evaluación de los modelos de Pythia entrenados usando ROUGE-L como métrica y teniendo en cuenta si la ontología se incluye en el prompt.

Property	Llama 3	Pythia Models													
	8B	12B		6.9B		2.8B		1.4B		1B		410m		160m	
	Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.
Mission Name	0.924	0.843	0.864	0.781	0.820	0.804	0.845	0.695	0.764	0.636	0.810	0.732	0.694	0.330	0.135
Mission Status	0.515	0.552	0.563	0.472	0.562	0.516	0.603	0.464	0.484	0.389	0.557	0.455	0.361	0.061	0.020
Launch Date	0.668	0.639	0.670	0.600	0.652	0.627	0.661	0.534	0.597	0.502	0.571	0.566	0.377	0.029	0.000
EOL Date	0.226	0.260	0.301	0.206	0.271	0.207	0.241	0.251	0.253	0.214	0.258	0.165	0.111	0.000	0.000
Objectives	0.156	0.136	0.139	0.122	0.140	0.136	0.143	0.127	0.138	0.115	0.136	0.134	0.127	0.059	0.028
Orbit Inclination	0.607	0.529	0.513	0.481	0.523	0.544	0.568	0.444	0.529	0.489	0.547	0.485	0.522	0.259	0.065
Orbit Altitude	0.759	0.619	0.607	0.535	0.581	0.633	0.633	0.540	0.592	0.538	0.606	0.595	0.595	0.311	0.115
Orbit Type	0.742	0.506	0.496	0.484	0.504	0.557	0.508	0.477	0.503	0.467	0.549	0.519	0.526	0.142	0.110
Instrument Name	0.460	0.513	0.533	0.526	0.531	0.493	0.540	0.478	0.432	0.433	0.507	0.503	0.381	0.407	0.276
Instrument MeasurementsApp	0.160	0.154	0.160	0.178	0.161	0.149	0.148	0.152	0.153	0.138	0.157	0.155	0.171	0.122	0.187
Stakeholder Name	0.413	0.385	0.416	0.470	0.441	0.400	0.468	0.339	0.400	0.500	0.390	0.437	0.441	0.256	0.241
Average	0.512	0.467	0.478	0.442	0.471	0.461	0.487	0.409	0.440	0.402	0.463	0.432	0.391	0.180	0.107

Tabla 5. Evaluación de los modelos de Pythia entrenados usando la similitud basada en LLM como métrica y teniendo en cuenta si la ontología se incluye en el prompt.

Property	Llama 3	Pythia Models													
	8B	12B		6.9B		2.8B		1.4B		1B		410m		160m	
	Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.	Ont.	No Ont.
Mission Name	0.949	0.851	0.902	0.790	0.876	0.823	0.894	0.718	0.826	0.655	0.876	0.766	0.769	0.372	0.364
Mission Status	0.685	0.678	0.657	0.542	0.671	0.639	0.691	0.554	0.606	0.458	0.653	0.559	0.512	0.071	0.037
Launch Date	0.738	0.666	0.704	0.640	0.661	0.661	0.698	0.556	0.602	0.532	0.635	0.562	0.345	0.029	0.000
EOL Date	0.325	0.299	0.342	0.261	0.349	0.284	0.311	0.273	0.293	0.250	0.349	0.218	0.162	0.000	0.000
Objectives	0.669	0.573	0.622	0.507	0.624	0.538	0.616	0.480	0.621	0.444	0.623	0.527	0.643	0.260	0.697
Orbit Inclination	0.795	0.643	0.628	0.572	0.636	0.664	0.674	0.536	0.630	0.575	0.653	0.605	0.614	0.351	0.088
Orbit Altitude	0.786	0.630	0.637	0.575	0.609	0.663	0.661	0.563	0.617	0.561	0.634	0.618	0.623	0.343	0.138
Orbit Type	0.677	0.514	0.520	0.474	0.525	0.550	0.530	0.453	0.535	0.450	0.558	0.511	0.550	0.153	0.195
Instrument Name	0.627	0.647	0.628	0.646	0.644	0.629	0.687	0.608	0.593	0.587	0.641	0.644	0.580	0.525	0.423
Instrument MeasurementsApp	0.696	0.655	0.678	0.666	0.645	0.657	0.676	0.620	0.657	0.620	0.645	0.676	0.624	0.611	0.617
Stakeholder Name	0.561	0.538	0.559	0.572	0.569	0.536	0.581	0.500	0.526	0.618	0.523	0.585	0.564	0.431	0.451
Average	0.683	0.609	0.625	0.568	0.619	0.604	0.638	0.533	0.591	0.523	0.617	0.570	0.544	0.286	0.274

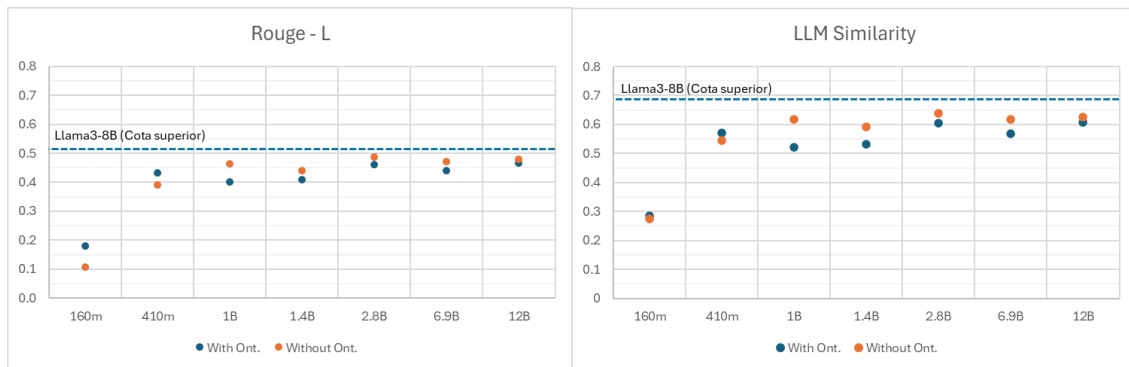


Figura 6. Comparación de la evaluación de los modelos Pythia entrenados incluyendo o no la ontología en el prompt de acuerdo con las métricas Rouge-L y la similitud calculada usando un LLM.

Para facilitar el análisis de los resultados de la evaluación semántica de los modelos se presenta la Figura 6 donde se incluyen los valores promedios de Rouge-L y la similitud basada en LLM para todas las propiedades. Nuevamente se utiliza Llama3-8B (zero-shot) como línea base, aunque hay que considerar que el número de Turtle validos generados por este modelo es considerablemente menor que el que generan los modelos ajustados (Ver Figura 5).

En la figura se observa que a partir de 1B de parámetros siempre es mejor excluir la ontología del prompt. Esto es un beneficio pues un prompt más corto se procesa más rápido y requiere menos memoria de GPU lo que se traduce en un ahorro de costes. Teniendo en cuenta las dos métricas el mejor modelo ajustado es el modelo de 2.8B sin incluir la ontología en el prompt. Es decir, en esta tarea se ha demostrado que con solo 2.8B parámetros se obtiene un modelo de mejor desempeño que modelos mucho más grandes como los modelos de 6.9B y 12B. Nuevamente, un modelo de menos parámetros se ejecuta más rápido y reduce costos relacionados con la infraestructura requerida para inferencia. Incluso, el modelo de 1B de parámetros es competitivo y comparable con los modelos de 6.9B y 12B de parámetros.

7 Conclusiones

En este entregable se presenta el proceso que se ha seguido para ajustar LLMs para generar instancias de una ontología en formato Turtle. Se usan los modelos Pythia porque permiten identificar que tamaño de modelo es el más adecuado de un abanico de 8 modelos entre 14M y 12B de parámetros entrenados en el mismo corpus y bajo las mismas condiciones. Aunque estos modelos han visto en fase de entrenamiento ontologías probablemente almacenadas en GitHub, se comprobó que para una ontología que no han visto previamente en entrenamiento no son capaces de extraer instancias y formalizarlas en Turtle.

Para los experimentos se selecciona el dominio de las misiones espaciales debido a la existencia de información textual e información estructurada acerca de las misiones que es útil tanto para el entrenamiento como la evaluación de los modelos ajustados. Con la ayuda Llama3-8B y una ontología de misiones espaciales se genera un conjunto de datos de entrenamiento y validación.

Los resultados del entrenamiento y de la evaluación indican que modelos a partir de 410M de parámetros aprenden a realizar la tarea de extraer instancias de una ontología a partir de un texto y formalizarla en Turtle. El modelo de 410M genera Turtle valido sintácticamente para al menos el 78% de las misiones. Este porcentaje se eleva al 90% para los modelos de 1B o más de parámetros.

Además, si analizamos las instancias extraídas con respecto a la base de datos de referencia (la base de datos CEOS) encontramos que el mejor modelo ajustado es el de 2.9B de parámetros que supera a modelos muchos más grandes de 6.9B y 12B de parámetros. Este resultado puede estar relacionado al tamaño del conjunto de datos de entrenamiento que podría no ser lo suficientemente grande para ajustar mejor los modelos más grandes. De hecho, el modelo de 1B de parámetros es competitivo y casi a la par que los modelos de 6.9B y 12B de parámetros. Este es un resultado importante porque modelos más pequeños son más eficientes en inferencia y requieren menos memoria de GPU lo que se traduce en menor costo.

Otro resultado reportado es que para modelos de 1B o más de parámetros incluir la ontología en el prompt produce peores resultados en la evaluación semántica. La no inclusión de la ontología en el prompt permite trabajar con prompts de menor tamaño que tienen un impacto positivo tanto en el rendimiento de los modelos en tiempo de inferencia y el uso de memoria de GPU.

8 Referencias

Berquand, A., Ladeira A. V. (2022). From Mission Description to Knowledge Graph: Applying Transformer-based models to map knowledge from publicly available satellite datasets

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (pp. 1148-1158).

Surdeanu, M., & Ji, H. (2014). Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In Proc. Text Analysis Conference (TAC2014).

Wu, L., Petroni, F., Josifoski, M., Riedel, S., & Zettlemoyer, L. (2019). Scalable zero-shot entity linking with dense entity retrieval. arXiv preprint arXiv:1911.03814.

Apéndice A: Ontología de misiones espaciales

A continuación, se presenta el código Turtle de la ontología de misiones espaciales. Además, al final se incluye instancias de la ontología a manera de ejemplo.

```
@prefix : <http://ontologies.org/spaceont#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix spaceont: <http://ontologies.org/spaceont#> .
@base <http://ontologies.org/spaceont#> .

<http://ontologies.org/spaceont#> rdf:type owl:Ontology .

#####
#      Object Properties
#####

### http://ontologies.org/spaceont#collaboratesWith
spaceont:collaboratesWith rdf:type owl:ObjectProperty ;
                           rdfs:subPropertyOf owl:topObjectProperty ;
                           rdfs:domain spaceont:Stakeholder ;
                           rdfs:range spaceont:Stakeholder ;
                           rdfs:comment "A Stakeholder collaborates with
another Stakeholder."@en ;
                           rdfs:label "collaboratesWith"@en .

### http://ontologies.org/spaceont#enables
spaceont:enables rdf:type owl:ObjectProperty ;
                  rdfs:subPropertyOf owl:topObjectProperty ;
                  rdfs:domain spaceont:Instrument ,
                              spaceont:Mission ;
                  rdfs:range spaceont:Application ;
                  rdfs:comment "An Instrument or a Mission, through its
instruments, enables an Application."@en ;
                  rdfs:label "enables"@en .

### http://ontologies.org/spaceont#hasInstrument
spaceont:hasInstrument rdf:type owl:ObjectProperty ;
                       rdfs:subPropertyOf owl:topObjectProperty ;
                       rdfs:domain spaceont:Mission ;
                       rdfs:range spaceont:Instrument ;
                       rdfs:comment "A Mission has an Instrument"@en ;
                       rdfs:label "hasInstrument"@en .

### http://ontologies.org/spaceont#hasOrbit
spaceont:hasOrbit rdf:type owl:ObjectProperty ;
                  rdfs:subPropertyOf owl:topObjectProperty ;
                  rdfs:domain spaceont:Mission ;
                  rdfs:range spaceont:Orbit ;
                  rdfs:comment "A Mission has an Orbit"@en ;
                  rdfs:label "hasOrbit"@en .

### http://ontologies.org/spaceont#isBasedIn
spaceont:isBasedIn rdf:type owl:ObjectProperty ;
                   rdfs:subPropertyOf owl:topObjectProperty ;
                   rdfs:domain spaceont:Stakeholder ;
                   rdfs:range spaceont:Country ;
```

```

rdfs:comment "A Stakeholder is based in a Country"@en ;
rdfs:label "isBasedIn"@en .

### http://ontologies.org/spaceont#isPrimeOf
spaceont:isPrimeOf rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain spaceont:Stakeholder ;
rdfs:range spaceont:Mission ;
rdfs:comment "A Stakeholder is the main stakeholder of a
Mission: <Stakeholder isPrimeOf Mission>"@en ;
rdfs:label "isPrimeOf"@en .

### http://ontologies.org/spaceont#managesMission
spaceont:managesMission rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain spaceont:Stakeholder ;
rdfs:range spaceont:Mission ;
rdfs:comment "A stakeholder manages a Mission"@en ;
rdfs:label "managesMission"@en .

### http://ontologies.org/spaceont#ownsInstrument
spaceont:ownsInstrument rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain spaceont:Stakeholder ;
rdfs:range spaceont:Instrument ;
rdfs:comment "A Stakeholder owns an Instrument"@en ;
rdfs:label "ownsInstrument"@en .

### http://ontologies.org/spaceont#subStakeholderOf
spaceont:subStakeholderOf rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf owl:topObjectProperty ;
rdfs:domain spaceont:Stakeholder ;
rdfs:range spaceont:Stakeholder ;
rdfs:comment "A Stakeholder is a subsidiary of
another Stakeholder" ;
rdfs:label "subStakeholderOf"@en .

#####
# Data properties
#####

### http://ontologies.org/spaceont#ApplicationType
spaceont:ApplicationType rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Application ;
rdfs:range xsd:string ;
rdfs:comment "Type of Application."@en ;
rdfs:label "applicationType"@en .

### http://ontologies.org/spaceont#InstrumentName
spaceont:InstrumentName rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Instrument ;
rdfs:range xsd:string ;
rdfs:comment "Name of the instrument"@en ;
rdfs:label "InstrumentName"@en .

### http://ontologies.org/spaceont#InstrumentStatus

```

```
spaceont:InstrumentStatus rdf:type owl:DatatypeProperty ;
                           rdfs:subPropertyOf owl:topDataProperty ;
                           rdfs:domain spaceont:Instrument ;
                           rdfs:range xsd:string ;
                           rdfs:comment "Status of the instrument, for example,
Proposed, Approved, Being developed, Operational, No longer operational."@en ;
                           rdfs:label "InstrumentStatus"@en .
```

```
### http://ontologies.org/spaceont#InstrumentType
spaceont:InstrumentType rdf:type owl:DatatypeProperty ;
                           rdfs:subPropertyOf owl:topDataProperty ;
                           rdfs:domain spaceont:Instrument ;
                           rdfs:range xsd:string ;
                           rdfs:comment "Type of an instrument. For example,
Lidars, Atmospheric chemistry, and Imaging microwave radars"@en ;
                           rdfs:label "InstrumentType"@en .
```

```
### http://ontologies.org/spaceont#OrbitInclination
spaceont:OrbitInclination rdf:type owl:DatatypeProperty ;
                           rdfs:subPropertyOf owl:topDataProperty ;
                           rdfs:domain spaceont:Orbit ;
                           rdfs:range xsd:string ;
                           rdfs:comment "Orbit inclination in degrees."@en ;
                           rdfs:label "orbitInclination"@en .
```

```
### http://ontologies.org/spaceont#countryName
spaceont:countryName rdf:type owl:DatatypeProperty ;
                       rdfs:subPropertyOf owl:topDataProperty ;
                       rdfs:domain spaceont:Country ;
                       rdfs:range xsd:string ;
                       rdfs:comment "Name of a country, e.g., Spain, France, The
United States, Japan, etc."@en ;
                       rdfs:label "countryName"@en .
```

```
### http://ontologies.org/spaceont#endOfLife
spaceont:endOfLife rdf:type owl:DatatypeProperty ;
                    rdfs:subPropertyOf owl:topDataProperty ;
                    rdfs:domain spaceont:Mission ;
                    rdfs:range xsd:dateTime ;
                    rdfs:comment "End of life date"@en ;
                    rdfs:label "endOfLife"@en .
```

```
### http://ontologies.org/spaceont#launchDate
spaceont:launchDate rdf:type owl:DatatypeProperty ;
                     rdfs:subPropertyOf owl:topDataProperty ;
                     rdfs:domain spaceont:Mission ;
                     rdfs:range xsd:dateTime ;
                     rdfs:comment "Mission launch date"@en ;
                     rdfs:label "launchDate"@en .
```

```
### http://ontologies.org/spaceont#measurementsApp
spaceont:measurementsApp rdf:type owl:DatatypeProperty ;
                           rdfs:subPropertyOf owl:topDataProperty ;
                           rdfs:domain spaceont:Instrument ;
                           rdfs:range xsd:string ;
                           rdfs:comment "Measurements and applications of an
instrument. An example is \"Nadir viewing sounding radar for provision of
real-time high-precision sea surface topography, ocean circulation and wave
height data.\""@en ;
```

```

rdfs:label "measurementsApp"@en .

### http://ontologies.org/spaceont#missionName
spaceont:missionName rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Mission ;
rdfs:range xsd:string ;
rdfs:comment "Name of the mission"@en ;
rdfs:label "missionName"@en .

### http://ontologies.org/spaceont#missionStatus
spaceont:missionStatus rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Mission ;
rdfs:range xsd:string ;
rdfs:comment "Mission status including Considered,
Approved, Planned, Operational,Operational (extended), Operational (nominal),
Commissioning,"@en ;
rdfs:label "missionStatus"@en .

### http://ontologies.org/spaceont#objectives
spaceont:objectives rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Mission ;
rdfs:range xsd:string ;
rdfs:comment "Objective of the mission or what the mission
aims at. For instance, the objective of the AIRMO Constellation is:
Monitor GHG emissions, CH4, in various and often remote locations. Aims
to detect every super emitter, particularly in the Oil & Gas industry, by
2030"@en ;
rdfs:label "objectives"@en .

### http://ontologies.org/spaceont#orbitAltitud
spaceont:orbitAltitud rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Orbit ;
rdfs:range xsd:string ;
rdfs:comment "Orbit Altitud in kilometers km."@en ;
rdfs:label "orbitAltitud"@en .

### http://ontologies.org/spaceont#orbitRepeatCycle
spaceont:orbitRepeatCycle rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Orbit ;
rdfs:range xsd:string ;
rdfs:comment "Orbit repeat cycle, usually defined in
days."@en ;
rdfs:label "orbitRepeatCycle"@en .

### http://ontologies.org/spaceont#orbitType
spaceont:orbitType rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf owl:topDataProperty ;
rdfs:domain spaceont:Orbit ;
rdfs:range xsd:string ;
rdfs:comment "Orbit type. For instance Sun-synchronous,
Highly elliptical, Inclined, non-sun-synchronous"@en ;
rdfs:label "orbitType"@en .

```

```

### http://ontologies.org/spaceont#program
spaceont:program rdf:type owl:DatatypeProperty ;
                  rdfs:subPropertyOf owl:topDataProperty ;
                  rdfs:domain spaceont:Mission ;
                  rdfs:range xsd:string ;
                  rdfs:comment "A program refers to a budget where works are
grouped. Examples of programs are:Copernicus Programme, Cosmic Vision,
ExoMars, Launchers, Earth Observation"@en ;
                  rdfs:label "program"@en .

### http://ontologies.org/spaceont#stakeholderName
spaceont:stakeholderName rdf:type owl:DatatypeProperty ;
                        rdfs:subPropertyOf owl:topDataProperty ;
                        rdfs:domain spaceont:Stakeholder ;
                        rdfs:range xsd:string ;
                        rdfs:comment "Name of the stakeholder"@en ;
                        rdfs:label "stakeholderName"@en .

#####
#   Classes
#####

### http://ontologies.org/spaceont#Application
spaceont:Application rdf:type owl:Class ;
                    rdfs:comment "Text describing how the space mission
results or instruments can be used in certain domains or task. For example,
the following text is an application of an instrument: SAR for land
applications of cartography and disaster monitoring." ;
                    rdfs:label "Application"@en .

### http://ontologies.org/spaceont#Country
spaceont:Country rdf:type owl:Class ;
                rdfs:comment "A country is a distinct part of the world, such
as a state, nation, or other political entity. For instance, Spain, Germany,
or France are a country."@en ;
                rdfs:label "Country"@en .

### http://ontologies.org/spaceont#Instrument
spaceont:Instrument rdf:type owl:Class ;
                   rdfs:comment "Space instruments gather scientific
information to explore planets, moons and our solar system and beyond. They
can be categorized in Direct-sensing instruments, also called contact science
instruments, that register characteristics of phenomena in their immediate
vicinity, and Remote-sensing instruments that form some kind of image or
characterization of the source of the phenomena that enter the instrument." ;
                   rdfs:label "Instrument"@en .

### http://ontologies.org/spaceont#Mission
spaceont:Mission rdf:type owl:Class ;
                rdfs:subClassOf owl:Thing ;
                rdfs:comment "A journey, by a manned or unmanned vehicle,
into space for a specific reason (normally to gather scientific data)"@en ;
                rdfs:label "Mission"@en .

### http://ontologies.org/spaceont#Orbit
spaceont:Orbit rdf:type owl:Class ;
              rdfs:comment "An elliptical movement of an object about a
celestial object or Lagrange point, especially a periodic elliptical
revolution"@en ;

```

```

rdfs:label "Orbit"@en .

### http://ontologies.org/spaceont#Stakeholder
spaceont:Stakeholder rdfs:type owl:Class ;
                    rdfs:subClassOf owl:Thing ;
                    rdfs:comment "A person or organisation with a legitimate
interest in a given situation, action or enterprise"@en ;
                    rdfs:label "Stakeholder"@en .

#####
# Individuals
#####

### http://ontologies.org/spaceont#AATSR
spaceont:AATSR rdfs:type owl:NamedIndividual ,
                spaceont:Instrument ;
                spaceont:enables spaceont:fire_observation ,
                                spaceont:sea_surface_temperature ;
                spaceont:InstrumentName "Advanced Along Track Scanning
Radiometer" ;
                spaceont:InstrumentType "infrared radiometer" .

### http://ontologies.org/spaceont#ASAR
spaceont:ASAR rdfs:type owl:NamedIndividual ,
                spaceont:Instrument ;
                spaceont:InstrumentName "Advanced Synthetic Aperture Radar" .

### http://ontologies.org/spaceont#Astrium_Germany
spaceont:Astrium_Germany rdfs:type owl:NamedIndividual ,
                            spaceont:Stakeholder ;
                            spaceont:isBasedIn spaceont:Germany ;
                            spaceont:isPrimeOf spaceont:Envisat ;
                            spaceont:countryName "Astrium Germany" .

### http://ontologies.org/spaceont#Envisat
spaceont:Envisat rdfs:type owl:NamedIndividual ,
                    spaceont:Mission ;
                    spaceont:hasInstrument spaceont:AATSR ,
                                           spaceont:ASAR ,
                                           spaceont:MERIS ,
                                           spaceont:SCIAMACHY ;
                    spaceont:endOfLife "2012-05-09T00:00:00"^^xsd:dateTime ;
                    spaceont:launchDate "2002-03-01T00:00:00"^^xsd:dateTime ;
                    spaceont:missionName "Envisat" ;
                    spaceont:missionStatus "End" .

### http://ontologies.org/spaceont#Germany
spaceont:Germany rdfs:type owl:NamedIndividual ,
                    spaceont:Country ;
                    spaceont:countryName "Germany" ;
                    spaceont:endOfLife "2012-05-09"^^xsd:dateTime .

### http://ontologies.org/spaceont#MERIS
spaceont:MERIS rdfs:type owl:NamedIndividual ,
                spaceont:Instrument ;
                spaceont:InstrumentName "Medium Resolution Imaging
Spectrometer" ;
                spaceont:InstrumentType "maging spectrometer" .

```

```
### http://ontologies.org/spaceont#SCIAMACHY
spaceont:SCIAMACHY rdf:type owl:NamedIndividual ,
                    spaceont:Instrument ;
                    spaceont:enables spaceont:measurements_of_trace_gases ;
                    spaceont:InstrumentType "imaging spectrometer" .

### http://ontologies.org/spaceont#fire_observation
spaceont:fire_observation rdf:type owl:NamedIndividual ,
                               spaceont:Application ;
                               spaceont:ApplicationType "fire observation" .

### http://ontologies.org/spaceont#measurements_of_trace_gases
spaceont:measurements_of_trace_gases rdf:type owl:NamedIndividual ,
                                             spaceont:Application ;
                                             spaceont:ApplicationType "global
measurements of trace gases" .

### http://ontologies.org/spaceont#sea_surface_temperature
spaceont:sea_surface_temperature rdf:type owl:NamedIndividual ,
                                         spaceont:Application ;
                                         spaceont:ApplicationType "sea surface
temperature" .
```